Japanese Patent Application No. Hei 10[1998]-535431

Job No.: 228-119441
Ref.: Japanese pat. No. 547099/PF970008 US/JDH(Fideliz)/Order No. 8428
Translated from Japanese by the McElroy Translation Company
800-531-9977 customerservice@mcelroytranslation.com

[repeated header, will not be repeated in the following pages]
Delivery No. 547099
Delivery date: November 6, 2007

Non-Patent Exhibit

Patent application No.: Japanese Patent Application No. Hei 10[1998]-535431
Date of preparation: October 5, 2007
Preparing person: Takahiro Inose 9560 5G00
Title of the invention: Method for automatically adapting level of signal exchanged in communication network

----

[repeated header]

Book 1997-00176-001
Digital signal processing handbook
Edited by Denshi Tsushin Gakkai [IEICE]

Published by Omu Sha

Chapter 10. Adaptive signal processing

10-1 Basic theory

10-1-1 Introduction

A filter that has a so-called learning function, that is, a function with which system parameters (such as the impulse response) are successively estimated from the input signal and output signal (desired signal) of an unknown system. It is applied in an echo canceller, automatic equalizer, noise canceller, etc. [1]-[9]. Here, the portion of the adaptive filter for correcting the filter coefficient of the adaptive filter is called an adaptive algorithm.

Topics to be addressed for an adaptive algorithm include simultaneous realization of higher speed convergence, higher speed execution speed, and smaller scale hardware, etc.

However, in research on higher speed by using parallel processing, there are opposite requirements. Also, it is necessary to consider the stability of operation, and many researchers have proposed various schemes.

First, in 1960, in their research on adaptive switching circuits, Widrow and Hoff proposed an adaptive algorithm called the Widrow-Hoff least mean square algorithm (hereinafter to be referred to as LMS algorithm) [10]. For this algorithm, the filter coefficients are corrected so that the mean square error is minimized based on the deepest fall method. Because it has a small operation quantity, it still holds the position as the typical algorithm at present.

In 1967, independent of the aforementioned work, Noda and Nagumo proposed a learning (type) identification method [11], [12]. The learning identification method has better convergence characteristics although it has a little poorer performance with respect to the operation quantity than the aforementioned LMS algorithm. Consequently, it is an adaptive algorithm with excellent practical application. The learning identification method has another name, that is, the normalized LMS algorithm. It normalizes the coefficient correction item of the LMS algorithm by means of the state vector norm of the filter. Consequently, the learning identification method can also be deemed to be located on the extension line of the LMS algorithm. When it is used as an adaptive algorithm based on the orthogonal map theorem, highly interesting extension is possible.

In any case, the aforementioned LMS algorithm and the learning identification method are typical adaptive algorithms that support adaptive signal processing entering the stage of practical application. The aforementioned algorithms can be deemed to be repeating computation methods that can solve the Wiener-Hopf equation generated based on a statistical quantity of the signal even if the statistical properties of the given signal are unknown (or almost unknown). Also, even if the parameter to be estimated varies in a relatively slow way over time, it is still possible to track the variation in the parameter to a certain degree. However, it has been pointed out that when there is an input signal, said algorithms have a significantly diminished convergence speed. This is undesired.

On the other hand, in 1960, at the same time the LMS algorithm was published, Kalman proposed a discrete time Kalman filter [13]. It is believed that real time treatment with the algorithm proposed by Mr. Kalman is difficult even now. However, it is a famous theory that dramatically extends Wiener's idea. Also, for the Kalman filter, as an unknown parameter that requires estimation of a state variable, by assuming that the parameter does not vary over time (including irregular fluctuation), the Kalman filter agrees with the well known recursive least square algorithm (RLS) [14]. For the RLS algorithm, assuming that the number of parameters to be estimated is N, O (N^2) rounds of multiplication for each sample are necessary, so realization in hardware is difficult. However, when the aforementioned assumption stands, excellent

convergence characteristics are displayed. As a measure for coping with variation in an unknown parameter, the introduction of forgetting coefficient λ has been proposed. However, depending on the value of λ, numerical instability may take place. Consequently, the hitherto estimated value may jump to another value, or the algorithm may be unable to continue. Caution should be exercised. However, when the UD decomposition method is introduced into the RLS, such phenomenon can hardly take place. At the same time, effectiveness in forming hardware by means of a systolic array is also displayed [15].

Block signal processing based on the blocking of the input data and FFT has the excellent characteristic feature that the necessary number of rounds of multiplication operation can be reduced with respect to processing in the time zone. If such block signal processing is introduced into the filter portion of the adaptive filter, the adaptive algorithm also requires a fast processing matching it. In order to meet such requirement, G. A. Clark et al. [16] proposed BLMS (block LMS) algorithm [17-19]. Also, as a characteristic feature of block processing, studies have been performed on how to increase the processing speed of an adaptive filter by introducing the concept of parallel processing to filter computing and the adaptive algorithm. Also, the jump algorithm [27] has been proposed for increasing the convergence speed.

On the other hand, a plurality of input data has also been proposed to increase the convergence speed in input of colored signals instead of for increasing the processing speed, according to research performed by [illegible]moto, Maekawa [20], and Oseki, Umeda [21]. Their schemes are characterized by the fact that the estimated coefficient at a certain time gives the desired output power with respect to plural state vectors required in obtaining it. However, these schemes have the problem that the operation quantity is huge for each sample time. In order to reduce the operation quantity, Furukawa et al. introduced the concept of block processing as explained above in the processing described in References [20], [21] and proposed the BOP (block orthogonal projection) algorithm as its general representation [22]. A more specific algorithm is described in references [23], [24].

In the present chapter, if not specified otherwise, matrix A with dimensions (N x N) and vector B of (N x 1) are referred to as $A_{N,N}$ and B_N respectively.

10.1.2 Setting of topic and evaluation quantity

In the following, an explanation will be given regarding the topic setting and evaluation quantity needed for deriving the adaptive algorithm. In the following discussions, all signals will be sampled using an appropriate scheme, and the system as the object will be represented in a discrete time zone.

Before going to topic setting, first, the definition of error will be explained. It is believed that the following three types of errors exist.

(1) Output error As the most frequently used definition of error, as shown in Figure 10.1(a), difference z(k) - y(k), where z(k) represents the observation signal and y(k) represents the output of the estimation system.

(2) Input error As shown in Figure 10.1(b), this error is defined as the difference x(k) - y(k) in output y(k) of the estimation system, where x(k) represents the input signal, and z(k) represents the observation signal. In this case, the estimation system is taken as the reverse system estimation of the unknown system.

(3) Generalized error As shown in Figure 10.1(c), this error is defined as a combination of the aforementioned output error and input error. Especially, assuming that the transmission functions of estimation system 1 and estimation system 2 are A(z) and B(z), one has

$$\left.\begin{aligned} A(z) &= a_0 + a_1 z^{-1} + \cdots + a_m z^{-m} \\ B(z) &= 1 + b_1 z^{-1} + \cdots + b_n z^{-n} \end{aligned}\right\} \qquad (10 \cdot 1)$$

In this case, the generalized error e(k) is represented as follows:

$$\begin{aligned} e(k) = {} & z(k) + b_1 z(k-1) + \cdots + b_n z(k-n) \\ & - a_0 x(k) - a_1 x(k-1) - \cdots - a_m x(k-m) \end{aligned} \qquad (10 \cdot 2)$$

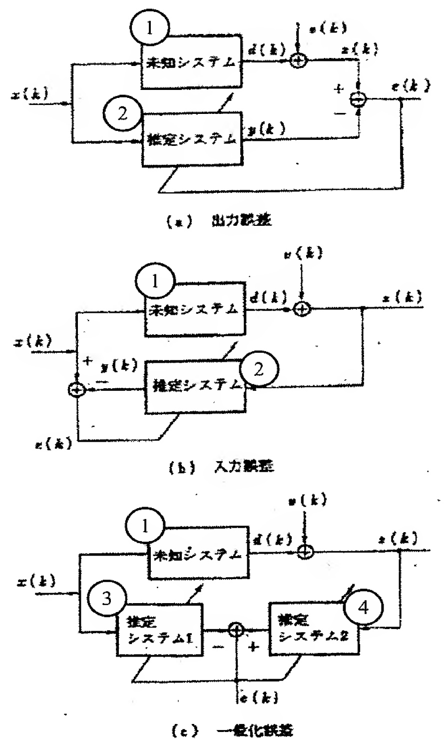


Figure 10.1. Definitions of errors.

Key: (a) Output error
(b) Input error

(c) Generalized error
1 Unknown system
2 Estimated system
3 Estimated system 1
4 Estimated system 2

Here, in order to make the explanation more concise, discussion is performed by constraining to an unknown system that outputs a signal defined below [x(k), $-\infty<k<\infty$]

$$d(k)=\sum_{i=0}^{M-1} w_i \cdot x(k-i) \qquad (10 \cdot 3)$$

with respect to the input signal [d(k), $-\infty< k<\infty$]. Here, k represents the sample No. (corresponding to the time); $w_0, w_1, \cdots, w_{M-1}$ represent the impulse response to be estimated, and M represents the number of impulse responses. In addition, x(k) represents the probability process. Here, when the impulse response sequence is presented in vector representation, one has

$$W_M=(w_0, w_1, \cdots, w_{M-1})^T \qquad (10 \cdot 4)$$

where, ^{T}: transposition of the vector.

On the other hand, consider another FIR digital filter having the following input/output relationship:

$$y(k)=\sum_{i=0}^{N-1} h_i \cdot x(k-i) \qquad (10 \cdot 5)$$

Here, h_i represents the ith filter coefficient (impulse response). Coefficient h_i may be rewritten to any value. Similarly, W_M represents

$$h_N=(h_0, h_1, \cdots, h_{N-1})^T \qquad (10 \cdot 6)$$

Here, when a certain evaluation quantity is defined concerning the "distance" between the aforementioned FIR digital filter and a parameter of the unknown system shown in formula (10.4), and coefficients h_i of the FIR digital filter are corrected so that the aforementioned evaluation quantity is minimized, this filter is called an adaptive filter, and the obtained coefficient h_i is called an estimated value.

Here, the topic is how to select evaluation quantity J. Here, as shown in Figure 10.1(a), J can be deemed to be the mean square of the output error. Figure 10.1(a) shows updating of the adaptive filter coefficient so that the mean square of the difference between observation signal z(k) and output y(k) of the estimation system becomes minimized. According to (a) in this figure, evaluation quantity J is given as follows:

$$\begin{aligned} J&=E[e^2(k)]=E[\{z(k)-y(k)\}^2] \\ &=E[\{(d(k)+v(k))-y(k)\}^2] \end{aligned} \qquad (10 \cdot 7)$$

In consideration of the objective of estimation of the parameter, it is important to take the "distance" between the unknown system and the estimation system as a direct evaluation quantity. However, because the parameter of the unknown system is unknown, it is impossible to

directly use the evaluation quantity, and the mean square of Equation (10.7) is often used as the evaluation quantity.

10.1.3 Formulation of the topic for parameter estimation

In the following, an explanation will be given regarding the foundation of parameter estimation and the general properties of the solution obtained in this case. As explained in section 10.1.2, the topic of parameter estimation can be formulated as a topic of minimization of the appropriately defined evaluation quantity. As shown in Equation (10.7), the evaluation quantity J is given by

$$J=E[e^2(k)]=E[\{(d(k)+v(k))-y(k)\}^2] \tag{10・8}$$

In Equation (10.8), by substituting the following input/output relationship of the adaptive filter

$$y(k)=\sum_{i=0}^{N-1} h_i\cdot x(k-i) \tag{10・9}$$

one can obtain the following secondary function pertaining to h_N in the matrix representation:

$$J=h_N^T A_{N,N}(k)h_N-2h_N^T v_N(k)+E[z^2(k)] \tag{10・10}$$

Here, one has

$$A_{N,N}(k)=E[x_N(k)x_N^T(k)] \tag{10・11}$$

$$v_N(k)=E[x_N(k)z(k)] \tag{10・12}$$

$$z(k)=d(k)+v(k) \tag{10・13}$$

$$d(k)=W_M^T x_N(k) \tag{10・14}$$

$$x_N(k)=(x(k),x(k-1),\cdots,x(k-N+1))^T \tag{10・15}$$

$$h_N=(h_0,h_1,\cdots,h_{N-1})^T \tag{10・16}$$

$$W_M=(w_0,w_1,\cdots,w_{N-1})^T \tag{10・17}$$

As can be seen from Equation (10.10), the topic of minimization of J becomes a so-called constraint-free optimization topic. In addition, it is well known that $A_{N,N}(k)$ is a positive constant value, and J is the most typical convex function pertaining to h_N, and it has a unique minimum value [25]. Here, optimum coefficient vector h_N for minimizing J at time k is represented as h_N(opt, k). Here, h_N(opt, k) is obtained by partial differentiation of both sides and setting the result at 0. That is, both sides of Equation (10.10) are partial differentiated with respect to h_N, and one obtains

$$\frac{\partial J}{\partial h_N}=2A_{N,N}(k)h_N-2v_N(k) \tag{10・18}$$

Consequently, one has

$$h_N(\mathrm{opt},k)=A_{N,N}^{-1}(k)v_N(k) \tag{10・19}$$

here, assuming that $A_{N,N}(k)$ has regularity. Equation (10.19) is called the Wiener-Hoff solution.

In the following, a brief explanation will be given regarding the relationship between true value vector W_M and $h_N(opt,k)$. Here, as the magnitude relationship between M and N, one has M > N. In this case, consider that the initial N components and the remaining components of the true value vector are divided as follows:

$$W_M = (\overbrace{w_0, w_1, \cdots w_{N-1}}^{F_N}, \underbrace{w_N, w_{N+1}, \cdots, w_{M-1}}_{R_{M-N}})^T \qquad (10 \cdot 20)$$

As a result, the output (observation signal) z(k) from the unknown system becomes:

$$z(k) = x_N^T(k) F_N + x_{M-N}^T(k-N) R_{M-N} + v(k) \qquad (10 \cdot 21)$$

Consequently, $v_N(k)$ can be transformed as follows:

$$\begin{aligned} v_N(k) &= E[x_N(k) \cdot x_N^T(k)] F_N \\ &\quad + E[x_N(k) \cdot \{x_{M-N}^T(k-N) R_{M-N} + v(k)\}] \\ &= A_{N,N}(k) F_N + E[x_N(k) \cdot \{x_{M-N}^T(k-N) \\ &\quad \cdot R_{M-N} + v(k)\}] \end{aligned} \qquad (10 \cdot 22)$$

In this case, $h_N(opt,k)$ is given by Equation (10.19) as follows:

$$h_N(\mathrm{opt}, k) = F_N + B_N(k) \qquad (10 \cdot 23)$$

Here, bias vector $B_N(k)$ is given as follows:

$$\begin{aligned} B_N(k) &= A_{N,N}^{-1}(k) \\ &\quad \cdot E[x_N(k) \cdot \{x_{M-N}^T(k-N) R_{M-N} + v(k)\}] \end{aligned} \qquad (10 \cdot 24)$$

As a specific case, it is assumed that

$$E[x_N(k) \cdot x_{M-N}^T(k-N)] = 0 \qquad (10 \cdot 25)$$

$$E[x_N(k) \cdot v(k)] = 0 \qquad (10 \cdot 26)$$

As a result, optimum coefficient vector n_N(opt, k) becomes:

$$h_N(\mathrm{opt}, k) = F_N \qquad (10 \cdot 27)$$

That is, independent of time, it is in agreement with the initial N components of true value vector W_M.

On the other hand, when $M \leq N$, bias vector $B_N(k)$ is given as

$$B_N(k) = A_{N,N}^{-1}(k) E[x_N^T(k) \cdot v(k)] \qquad (10 \cdot 28)$$

Consequently, if Equation (10.26) is established, the following relationship, like Equation (10.27) is obtained:

$$h_N(\mathrm{opt}, k) = W_N \qquad (10 \cdot 29)$$

Here, one has

$$W_N = (\overbrace{w_0, w_1, \cdots w_{M-1}}^{W_M}, \underbrace{0, 0, \cdots, 0}_{(N-M)個})^T \qquad (10 \cdot 30)$$

Even if the aforementioned Equations (10.25), (10.26) are not established in the strict sense, they usually still may be assumed in practical application. Also, when the influence of bias

vector $B_N(k)$ cannot be ignored, the following scheme may also be adopted: $B_N(k)$ is adaptively estimated and subtracted from the optimum coefficient vector.

The constitution for determining the optimum coefficient vector based on Equation (10.19) contains an averaging operation and computing of an inverted matrix as shown in Figure 10.2, so it is inappropriate for real time processing. Here, as a method for high efficiency successive performance of the aforementioned averaging operation and computing of the inverted matrix, the recurrence least square (hereinafter to be referred to as RLS) method [14, 26] has been proposed. In addition, because $A_{N,N}(k)$ is a Toeplitz matrix, several methods are known that can significantly improve the computing procedure of the RLS [15, 28]. Also, Equation (10.19) can be solved by means of a planning method (descending method).

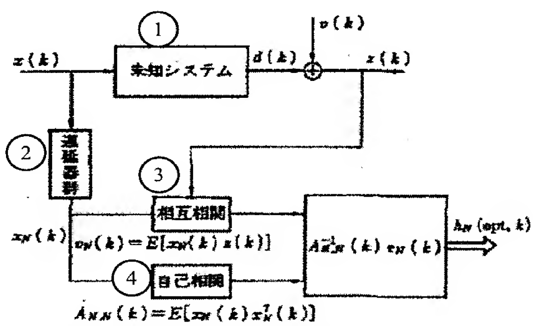


Figure 10.2. Constitution of Wiener-Hoff filter (assuming an ergodic property of the signal).

Key: 1 Unknown system
2 Group of delay units
3 Mutual correlation
4 Self correlation

The most typical method is the steepest descending method. Fast computing schemes belonging to them include the LMS algorithm [10], jump algorithm [27], etc. In addition, there are also the learning identification method and other computing schemes based on the orthogonal mapping theorem [11],[20-22].

In the following section, application examples of a typical adaptive algorithm and adaptive filter will be explained. Here, it is assumed that all signals to be handled in the following discussion are steady state probability processes, and an ergodic property is assumed.

10.2 Recurrence least square (RLS) method

In the following, an explanation will be given regarding derivation of the RLS. Here, instead of solving Equation (10.19) once, the RLS is determined by solving h_N(opt) while $A_{N,N}$

and v_N are determined using the recurrence method. Consequently, the estimated vector h_N is updated for each sample, so that h_N(opt) is gradually approached.

Here, by assuming that the signal has a steady state property and an ergodic property, the following quantity is defined:

$$X_{k-N+2,N}(k) = \begin{bmatrix} x(k) & x(k-1) & \cdots & x(k-N+1) \\ x(k-1) & x(k-2) & \cdots & x(k-N) \\ \vdots & & \cdots & \\ x(N-1) & x(N-2) & \cdots & x(0) \end{bmatrix} \tag{10・31}$$

As a result, the estimated values $A_{N,N}$, V_N of $A_{N,N}$ and v_N at time k become:

$$\bar{A}_{N,N}(k) = \frac{1}{k-N+2} X^T_{k-N+2,N}(k) X_{k-N+2,N}(k) \tag{10・32}$$

$$\bar{v}_N(k) = \frac{1}{k-N+2} X^T_{k-N+2,N}(k) z_{k-N+2}(k) \tag{10・33}$$

Here,

$$z_{k-N+2}(k) = (z(k), z(k-1), \cdots, z(N-1))^T \tag{10・34}$$

Here,

$$A_{N,N} = \bar{A}_{N,N}(k), \qquad k \to \infty \tag{10・35}$$

$$v_N = \bar{v}_N(k), \qquad k \to \infty \tag{10・36}$$

From Equation (10.19), one has

$$h_N(\text{opt}) = [X^T_{k-N+2,N}(k) X_{k-N+2,N}(k)]^{-1} \cdot X^T_{k-N+2,N}(k) z_{k-N+2}(k), \qquad k \to \infty \tag{10・37}$$

Consequently, it is possible to set the estimated vector of h_N using the data obtained until time k as $h_N(k+1)$:

$$h_N(k+1) = [X^T_{k-N+2,N}(k) X_{k-N+2,N}(k)]^{-1} \cdot X^T_{k-N+2,N}(k) z_{k-N+2}(k) \tag{10・38}$$

It is clear that when $k \to \infty$, Equation (10.38) agrees with h_N(opt).

The next problem is transformed to a recurrence computing representation in which the right side of Equation (10.38) is computed while the data that have been obtained are efficiently used and new data are added. For this purpose, the following quantity is defined:

$$P_{N,N}(k) = [X^T_{k-N+2,N}(k) X_{k-N+2,N}(k)]^{-1} \tag{10・39}$$

Because Equation (10.38) can be transformed as follows:

$$\begin{aligned} h_N(k+1) = {} & [X^T_{k-N+1,N}(k-1) X_{k-N+1,N}(k-1) \\ & + x^T_N(k) x_N(k)]^{-1} \\ & \cdot [X^T_{k-N+1,N}(k-1) z_{k-N+1}(k-1) \\ & + x_N(k) z(k)] \end{aligned} \tag{10・40}$$

so that from the well known formula of the inverted matrix

$$(A+BC)^{-1}=A^{-1}-A^{-1}B(I+CA^{-1}B)^{-1}CA^{-1} \tag{10・41}$$

and Equation (10.39), the first item becomes

$$\begin{aligned}&[X^T_{k-N+1,N}(k-1)X_{k-N+1,N}(k-1)+x_N^T(k)x_N(k)]^{-1}\\&=P_{N,N}(k-1)-\frac{P_{N,N}(k-1)x_N(k)}{1+x_N^T(k)P_{N,N}(k-1)x_N(k)}\\&\quad\cdot x_N^T(k)P_{N,N}(k-1)\\&=P_{N,N}(k-1)-k_N(k)x_N^T(k)P_{N,N}(k-1)\end{aligned} \tag{10・42}$$

Here, one has

$$k_N(k)=\frac{P_{N,N}(k-1)x_N(k)}{1+x_N^T(k)P_{N,N}(k-1)x_N(k)} \tag{10・43}$$

Here, because the left side of Equation (10.42) is $P_{N,N}(k)$, $P_{N,N}(k)$ can be obtained from $P_{N,N}(k - 1)$:

$$P_{N,N}(k)=P_{N,N}(k-1)-k_N(k)x_N^T(k)P_{N,N}(k-1) \tag{10・44}$$

Consequently, Equation (10.40) becomes

$$\begin{aligned}h_N(k+1)=&[P_{N,N}(k-1)\\&-k_N(k)x_N^T(k)P_{N,N}(k-1)]\\&\cdot[X^T_{k-N+1,N}(k-1)z_{k-N+1}(k-1)\\&+x_N(k)z(k)]\\=&P_{N,N}(k-1)X^T_{k-N+1,N}(k-1)z_{k-N+1}(k-1)\\&-k_N(k)x_N^T(k)P_{N,N}(k-1)\\&\cdot X^T_{k-N+1,N}(k-1)z_{k-N+1}(k-1)\\&+[I_{N,N}-k_N(k)x_N^T(k)]\\&\cdot P_{N,N}(k-1)x_N(k)z(k)\end{aligned} \tag{10・45}$$

In addition, as to be explained later, Equation (10.45) can be transformed to the following representation. First, from Equations (10.38), (10.39), one has

$$\begin{aligned}&P_{N,N}(k-1)X^T_{k-N+1,N}(k-1)z_{k-N+1}(k-1)\\&=[X^T_{k-N+1,N}(k-1)X_{k-N+1,N}(k-1)]^{-1}\\&\quad\cdot X^T_{k-N+1,N}(k-1)z_{k-N+1}(k-1)\\&=h_N(k)\end{aligned} \tag{10・46}$$

Also, from Equation (10.43), one has

$$\begin{aligned}&k_N(k)[1+x_N^T(k)P_{N,N}(k-1)x_N(k)]\\&=P_{N,N}(k-1)x_N(k)\end{aligned} \tag{10・47}$$

This can be transformed to

$$k_N(k)=[I_{N,N}-k_N(k)x_N^T(k)]P_{N,N}(k-1)x_N(k) \tag{10・48}$$

Here, $I_{N,N}$ represents a unit matrix having N rows and N columns. Consequently, from Equations (10.46), (10.48), the aforementioned Equation (10.45) can be transformed to the following recurrence computing system:

$$h_N(k+1)=h_N(k)+k_N(k)[z(k)-x_N^T(k)h_N(k)] \tag{10・49}$$

The aforementioned can be summarized, and the computing procedure for the RLS listed in Table 10.1 can be obtained.

Table 10.1. Computing procedure 1 for the RLS.

$$
\begin{aligned}
&k_N(k) = \frac{P_{N,N}(k-1)x_N(k)}{1+x_N^T(k)P_{N,N}(k-1)x_N(k)} \\
&P_{N,N}(k) = [I_{N,N} - k_N(k)x_N^T(k)]P_{N,N}(k-1) \\
&h_N(k+1) = h_N(k) + k_N(k) \\
&\qquad \cdot[z(k) - x_N^T(k)h_N(k)]
\end{aligned}
$$

The RLS listed in Table 10.1 may also be transformed as follows. First, for both sides of Equation (10.44), $I_N(k)$ is multiplied on the right, and by using Equation (10.48), one obtains

$$
\begin{aligned}
&P_{N,N}(k)x_N(k) \\
&\quad = P_{N,N}(k-1)x_N(k) \\
&\qquad - k_N(k)x_N^T(k)P_{N,N}(k-1)x_N(k) \\
&\quad = [I_{N,N} - k_N(k)x_N^T(k)]P_{N,N}(k-1)x_N(k) \\
&\quad = k_N(k)
\end{aligned}
\tag{10・50}
$$

$k_N(k)$ obtained using Equation (10.50) is substituted into the No. 3 step of the procedure listed in Table 10.1, and the No. 1 step of the procedure in Table 10.1 is substituted into the No. 2 step, and one obtains the procedure listed in Table 10.2.

Table 10.2. Computing procedure 2 for the RLS.

$$
\begin{aligned}
&P_{N,N}(k) = P_{N,N}(k-1) \\
&\qquad - \frac{P_{N,N}(k-1)x_N(k)x_N^T(k)P_{N,N}(k-1)}{1+x_N^T(k)P_{N,N}(k-1)x_N(k)} \\
&h_N(k+1) = h_N(k) + P_{N,N}(k)x_N(k) \\
&\qquad \cdot[z(k) - x_N^T(k)h_N(k)]
\end{aligned}
$$

When the procedure listed in Table 10.1 or Table 10.2 is executed, initial values $h_N(0)$, $P_{N,N}(0)$ have to be determined. They may be selected as follows:

$$
\left.
\begin{aligned}
&h_N(0) = \text{(arbitrary)} \\
&P_{N,N}(0) = cI_{N,N} \ \text{(C is a very large positive number)}
\end{aligned}
\right\}
\tag{10.51}
$$

For details, the reader is referred to References [14], [26].

10.3 LMS algorithm

Here, the LMS algorithm [5], [10] (hereinafter to be referred to as LMS) and the steepest descending method as its foundation will be explained. First, a brief account will be given on the steepest descending method.

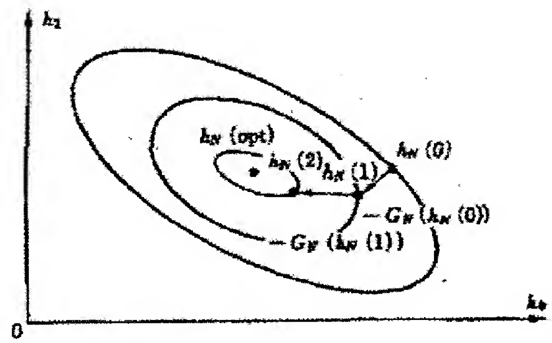


Figure 10.3. Contour lines of $E[e^2(k)]$,

At any h_N, the gradient vector $G_N(h_N)$ is defined as

$$G_N(h_N) = 2A_{N,N}h_N - 2v_N \qquad (10\cdot52)$$

(see Equation (10.18)). The secondary form of parameter h_N is present in Equation (10.10), and only one h_N corresponds to the minimum evaluation quantity J. Figure 10.3 shows the state when N = 2. The curves shown in Figure 10.3 correspond to collection of the contour of the values of J when coefficients h_0, h_1 vary. Here, $G_N(h_N)$ is equal to the gradient at any coefficient h_N, and it is in agreement with the direction of the normal to the contour lines. Consequently, by setting any point $h_N(0)$ as the initial value and moving $h_N(0)$ appropriately in the direction of - $G_N(h_N(0))$, J at $h_N(1)$ can be smaller than the value of J at $h_N(0)$. Here, $h_N(j - 1)$ represents the jth corrected value of h_N. By repeating this process, $h_N(j)$ infinitely approaches h_N(opt). The aforementioned algorithm can be summarized as follows:

$$h_N(j) = h_N(j-1) - 0.5\alpha(j)\, G_N(h_N(j-1)),$$
$$j = 1, 2, \cdots \qquad (10\cdot53)$$

Here, Equation (10.53) is called the steepest descending method, and $\alpha(j)$ is called the step gain. Here, for the aforementioned step gain, coefficient 0.5 is set for simplifying the later transformation of equations, and it does not have any special meaning. This constitution is shown in Figure 10.4.

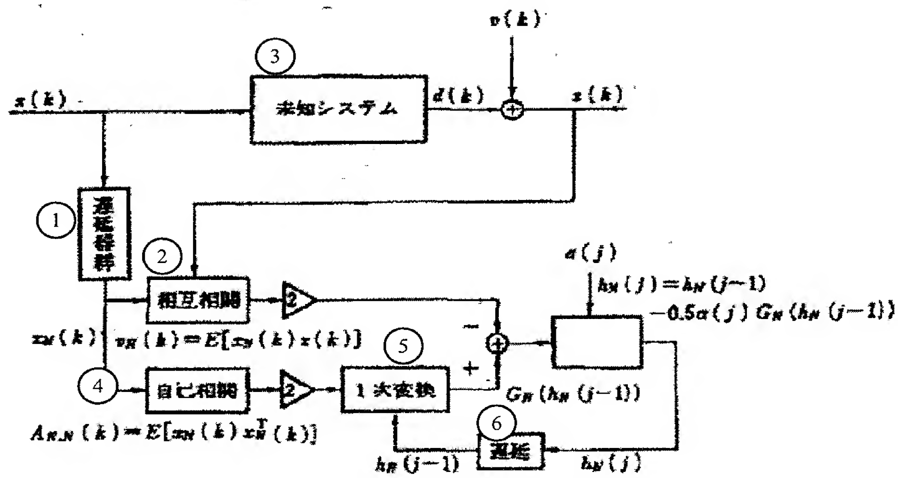


Figure 10.4. Constitution for determining the Wiener-Hoff solution using the steepest descending method.

Key:	1	Group of delay units
	2	Mutual correlation
	3	Unknown system
	4	Self correlation
	5	Primary conversion
	6	Delay

In the following, an explanation will be given regarding variation in the distance between $h_N(\text{opt})$ and $h_N(j)$ when h_N is corrected according to Equation (10.53). For this purpose, the error vector is defined as

$$E_N(j) = h_N(j) - h_N(\text{opt}) \qquad (10\cdot 54)$$

Here, from Equations (10.19), (10.52), and (10.53), the aforementioned Equation (10.54) can be transformed as follows.

$$\begin{aligned} E_N(j) &= [h_N(j-1) - 0.5\alpha(j)G_N(h_N(j-1))] \\ &\quad - h_N(\text{opt}) \\ &= (h_N(j-1) - h_N(\text{opt})) \\ &\quad - 0.5\alpha(j)G_N(h_N(j-1)) \\ &= E_N(j-1) - 0.5\alpha(j)(2A_{N,N}\cdot h_N(j-1) \\ &\quad - 2v_N) \\ &= E_N(j-1) - \alpha(j)A_{N,N}[h_N(j-1) \\ &\quad - A_{N,N}^{-1}\cdot v_N] \\ &= (I_{N,N} - \alpha(j)A_{N,N})E_N(j-1), \\ j &= 1, 2, 3, \cdots \end{aligned} \qquad (10\cdot 55)$$

Consequently, variation in the error vector $E_N(j)$ is given as

$$E_N(j) = \prod_{n=1}^{j}(I_{N,N} - \alpha(n)A_{N,N})\cdot E_N(0) \qquad (10\cdot 56)$$

It is known that whether the magnitude of $E_N(j)$ decreases with respect to an increase in j depends on step gain $\alpha(j)$ and $A_{N,N}$ (property of the input signal). In order to show the aforementioned property more clearly, $A_{N,N}$ is transformed as

$$A_{N,N} = Q_{N,N} \cdot D_{N,N} \cdot Q^T_{N,N} \qquad (10 \cdot 57)$$

Here, $Q_{N,N}$ represents an orthogonal matrix having the eigenvector of $A_{N,N}$ in the column vector:

$$Q^T_{N,N} = Q^{-1}_{N,N} \qquad (10 \cdot 58)$$

Also, $D_{N,N}$ is a diagonal matrix having the eigenvalues of $A_{N,N}$ as its diagonal elements.

$$D_{N,N} = \mathrm{Diag}(\lambda_1, \lambda_2, \cdots, \lambda_N), \quad \lambda_1 \le \lambda_2 \le \cdots \le \lambda_N \qquad (10 \cdot 59)$$

Consequently, one has

$$I_{N,N} - \alpha(j) A_{N,N} = Q_{N,N}[I_{N,N} - \alpha(j) D_{N,N}] Q^T_{N,N} \qquad (10 \cdot 60)$$

From Equations (10.56), (10.58) and (10.60), one has

$$E_N(j) = Q_{N,N} \prod_{n=1}^{j} (I_{N,N} - \alpha(n) D_{N,N}) \cdot Q^T_{N,N} \cdot E_N(0) \qquad (10 \cdot 61)$$

Here, when the value of $\alpha(j)$ is selected to be the reciprocal value of eigenvalue λ_j of $A_{N,N}$ for each correction, one has

$$\begin{aligned}\prod_{n=1}^{j} (I_{N,N} - \alpha(n) D_{N,N}) &= \prod_{n=1}^{j} (I_{N,N} - \lambda_n^{-1} \cdot D_{N,N}) \\ &= \mathrm{diag}(0, 1-\lambda_1^{-1}\lambda_2, \cdots, 1-\lambda_1^{-1}\lambda_N) \\ &\quad \cdot \mathrm{diag}(1-\lambda_2^{-1}\lambda_1, 0, 1-\lambda_2^{-1}\lambda_3, \cdots, 1-\lambda_2^{-1}\lambda_N) \\ &\quad \vdots \\ &\quad \cdot \mathrm{diag}(1-\lambda_N^{-1}\lambda_1, \cdots, 1-\lambda_N^{-1}\lambda_{N-1}, 0) \\ &= 0_{N,N} \qquad (10 \cdot 62)\end{aligned}$$

In the Nth and later corrections, $h_N(j)$ is in agreement with $h_N(opt)$. For example, when a white signal of appropriate magnitude is taken as the input signal, $A_{N,N}$ becomes a unit matrix. Consequently, assuming that $\alpha(j) = 1$ ($j = 1,2,...$), it is possible to obtain the optimum solution in a single round of correction. When the step gain is fixed without change each round, it is understood that the magnitude of the eigenvalue of $A_{N,N}$ can quickly approach the optimum solution uniformly. On the other hand, for an audio signal and other color signals, the ratio of the maximum eigenvalue to that of the minimum eigenvalue is very large, and, when the step gain is fixed, the convergence speed degrades significantly. Also, the property described here is retained to a certain degree even for the next LMS.

The above explanation is a discussion wherein the statistical quantity ($A_{N,N}$, v_N, or $G_N(h_N)$) is known. However, in practical application, usually, there is no time allowed to

compute the aforementioned statistical quantities. In the following, an explanation will be given regarding the LMS proposed by Widrow and Hoff.

When the averaging operation is omitted in Equation (10.53), Equation (10.53) can be transformed to

$$h_N(j) = h_N(j-1) - 0.5\alpha(j) \cdot [2x_N(k)x_N^T(k)h_N(j-1) - 2x_N(k)d(k)] \qquad (10 \cdot 63)$$

(see Equations (10.52), (10.11) and (10.12)). The LMS can be obtained from Equation 10.63 by assuming j = k + 1 and α(j) = α. That is,

$$\begin{aligned} h_N(k+1) &= h_N(k) - \alpha[y(k) - d(k)]x_N(k) \\ &= h_N(k) + \alpha e(k)x_N(k) \qquad (10 \cdot 64) \end{aligned}$$

In this way, from the data at time k, estimated vector h_N for subsequent use can be obtained repeatedly. Also, the selection of step gain α is determined from the statistical properties of the input signal as explained with regard to the steepest descending method. In the next α range

$$0 < \alpha < \frac{2}{\sum_{k=1}^{N} \lambda_k} \qquad (10 \cdot 65)$$

it is known that evaluation quantity $J = E[e^2(k)]$ approaches 0 [20]. Here, λ_N represents the maximum eigenvalue of $A_{N,N}$. Also, initial value $h_N(0)$ is at any point. This constitution is shown in Figure 10.5.

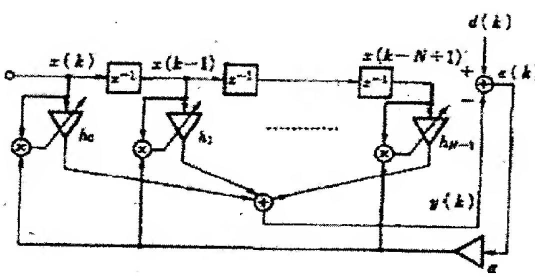


Figure 10.5. Constitution of LMS algorithm.

10.4 Learning identification method

In the following, an explanation will be given regarding the learning identification method proposed by Nagumo and Noda [11]. In the following discussion, if not specified otherwise, it is assumed that the unknown system and the known system have the same order number (M = N), and that no observation noise v(t) exists (v(t) = 0).

Now, consider the case when at time k, output y(k) of the adaptive filter is equal to output d(k) of the unknown system. That is, one has

$$d(k) = [x(k), x(k-1), \cdots, x(k-N+1)] \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{N-1} \end{bmatrix} \quad (10 \cdot 66)$$

It is clear that parameter W_N of the unknown system (true value vector) satisfies equation (10.66). However, parameter h_N (estimated vector) of the adaptive filter may not necessarily be equal to W_N. When Equation (10.66) is established for all input signals, $h_N = W_N$. In this way, h_N that satisfies equation (10.65) becomes a collection of all solutions including the true value vector. Here, as shown in Figure 10.6, the representative vector $h_N(k + 1)$ of h_N that satisfies equation (10.66) is taken as the foot of the perpendicular line drawn from an appropriately defined arbitrary point to the collection of solutions. As can be seen from Equation (10.66), the solution collection is perpendicular to state vector $x_N(k)$. In addition, because W_N is contained in the aforementioned solution collection, $h_N(k + 1)$ is the point nearest W_N when the coefficient is corrected from a certain point in the direction towards $x_N(k)$.

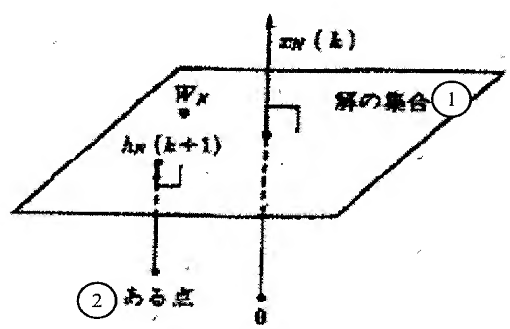


Figure 10.6. Solution collection.

Key: 1 Solution collection
2 Certain point

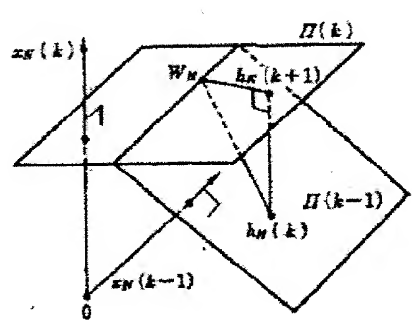


Figure 10.7. Geometric relationship of solutions obtained using the learning identification method.

In order to repeat the aforementioned operation such that $h_N(k + 1)$ approaches W_N, one may take $h_N(k)$ nearer to W_N than an appropriately determined point as the initial value of the following coefficient correction. This is shown in Figure 10.7. In Figure 10.7, $\Pi(k - 1)$ and $\Pi(k)$ represent solution collections at times k - 1 and k, respectively. In other words, they are collections of adaptive filter coefficients equal to desired signals d(k - 1), d(k) at times k - 1 and k, respectively. Also, because solution vector W_N refers to the point that becomes the desired signal d(k) with respect to all state vectors $[x_N(k) \quad (-\infty<k<\infty)]$, it is positioned at the crossing point of all solution collections $[\Pi(k) \quad (-\infty<k<\infty)]$.

The aforementioned can be summarized as follows:

$$\begin{aligned} h_N(k+1) &= h_N(k) + \{h_N(k+1) - h_N(k)\} \\ &= h_N(k) + \underbrace{\frac{\{W_N - h_N(k)\}^T\{h_N(k+1) - h_N(k)\}}{\|h_N(k+1) - h_N(k)\|}}_{\text{修正量 (a)}} \cdot \underbrace{\frac{h_N(k+1) - h_N(k)}{\|h_N(k+1) - h_N(k)\|}}_{\text{修正方向 (b)}} \end{aligned} \quad (10 \cdot 67)$$

Key: a Correction quantity
b Correction direction

Here, $\|\cdot\|$ represents the Euclid norm of the vector, and it is defined as the square root of the sum of the squares of the elements. Here, because

$$\frac{h_N(k+1) - h_N(k)}{\|h_N(k+1) - h_N(k)\|} = \frac{x_N(k)}{\|x_N(k)\|} \quad (10 \cdot 68)$$

$$\begin{aligned} \{W_N - h_N(k)\}^T x_N(k) &= d(k) - y(k) \\ &= e(k) \end{aligned} \quad (10 \cdot 69)$$

one may transform Equation (10.67) to

$$h_N(k+1) = h_N(k) + \frac{x_N(k)}{\|x_N(k)\|^2} e(k) \quad (10 \cdot 70)$$

The learning identification method can be realized by multiplying the correction vector of Equation (10.70) by the step gain to obtain

$$h_N(k+1) = h_N(k) + \alpha \frac{x_N(k)}{\|x_N(k)\|^2} e(k) \quad (10 \cdot 71)$$

If said assumption $(M=N, \; v(k)=0)$ is established, adaptive filter coefficient $h_N(k + 1)$ updated by means of Equation (10.70) gives the desired signal with respect to state vector $x_N(k)$ required for updating and independent of the initial value. In addition to the aforementioned

characteristic feature, even if the parameter of the unknown system varies, the aforementioned characteristic feature can still be retained if it is constrained to time k.

10.5 Block adaptive algorithm

In the following, an explanation will be given regarding the BOP algorithm (hereinafter to be referred to as BOP) that has the concept of the block treatment introduced to it in the learning identification method described in Section 10.4 and the BLMS algorithm (hereinafter to be referred to as BLMS) in the frequency zone.

10.5.1 BOP algorithm

In the following discussion, if not specified otherwise, it is assumed that the unknown system and the known system have the same order (M = N), and that no observation noise v(t) exists (v(t) = 0). The equation corresponding to Equation (10.66) is

$$X_{rN}(k)\cdot h_N = d_r(k) \qquad (10\cdot 72)$$

Here, at time k, state matrix $X_{rN}(k)$ and desired signal vector $d_r(k)$ are given as follows:

$$X_{rN}(k) = \begin{bmatrix} x(k) & x(k-1) & \cdots & x(k-N+1) \\ x(k-1) & x(k-2) & \cdots & x(k-N) \\ \vdots & & \ddots & \vdots \\ x(k-r+1) & x(k-r+2) & \cdots & x(k-r-N+2) \end{bmatrix} \qquad (10\cdot 73)$$

$$d_r(k) = [d(k), d(k-1), \cdots, d(k-r+1)]^T \qquad (10\cdot 74)$$

Here, r represents the quantity known as the block length. The treatment is performed with this block length as a unit. Also, there is a precondition that the block length be not over the size of the coefficient vector. That is,

$$r \leq N \qquad (10\cdot 75)$$

According to the BOP algorithm, coefficient correction is performed for every r samples. Consequently, compared with a method in which correction is performed for each sample, the computing quantity for each sample becomes 1/r.

In addition, for block Nos. L - 1 and L, one has

$$X_{rN}\{(L-1)r\}h_N^{(L)} = d_r\{(L-1)r\} \qquad (10\cdot 76)$$

$$X_{rN}(Lr)h_N^{(L+1)} = d_r(Lr) \qquad (10\cdot 77)$$

Here, $h_N^{(L)}$, $h_N^{(L+1)}$ represent any vectors in the solution spaces of block Nos. L - 1 and L, respectively (see Section 10.4). Also, assuming that the solution spaces of block Nos. L - 1 and L

are $\Pi^{(L-1)}$, $\Pi^{(L)}$, they are orthogonal to the spaces $S^{(L-1)}$, $S^{(L)}$ where the column vectors of the state matrix $X_{rN}\{(L-1)r\}$, $X_{rN}(Lr)$ are set. Consequently, although a little abstract, in Figure 10.7, one may read as follows:

$$\begin{aligned} X_N(k-1) &\rightarrow S^{(L-1)} \\ X_N(k) &\rightarrow S^{(L)} \\ h_N(k) &\rightarrow h_N^{(L)} \\ h_N(k+1) &\rightarrow h_N^{(L+1)} \end{aligned} \qquad (10 \cdot 78)$$

and one can derive the learning identification method of the block adaptive algorithm edition. Assuming that the adaptive filter coefficient used in the Lth block is $h_N^{(L)}$, from Equation (10.77), one has

$$X_{rN}(Lr)\{h_N^{(L+1)}-h_N^{(L)}\}=e_r(Lr) \qquad (10 \cdot 79)$$

Here, $e_r(Lr)$ represents the difference between desired signal vector $d_r(Lr)$ at the output error vector and output signal vector $y_r(Lr)$.

$$\begin{aligned} e_r(Lr) &= d_r(Lr)-y_r(Lr) \\ y_r(Lr) &= (y(Lr), y(Lr-1), \cdots, y(Lr-r+1))^T \end{aligned} \qquad (10 \cdot 80)$$

Numerous $h_N^{(L+1)}$ exist that satisfy Equation (10.79). However, for the points that make orthogonal mapping from $h_N^{(L)}$ to solution space $\Pi(L)$, $\| h_N^{(L+1)}-h_N^{(L)} \|$ is the smallest. Consequently, $h_N^{(L+1)}$ becomes the solution of the restrained associated equations:

$$\left.\begin{aligned} X_{rN}(Lr)\{h_N^{(L+1)}-h_N^{(L)}\}=e_r(Lr) \\ \min \| h_N^{(L+1)}-h_N^{(L)} \| \end{aligned}\right\} \qquad (10 \cdot 81)$$

Usually, the solution of Equation (10.81) can be represented as follows by using the general inverted matrix of Moore-Penrose:

$$h_N^{(L+1)}-h_N^{(L)}=X_{rN}^{+}(Lr)\,e_r(Lr) \qquad (10 \cdot 82)$$

Here, $X_{rN}^{+}(Lr)$ is the general inverted matrix of the Moore-Penrose form of $X_{rN}(Lr)$. BOP is given by applying a step gain to Equation (10.82):

$$h_N^{(L+1)}=h_N^{(L)}+\alpha X_{rN}^{+}(Lr)\,e_r(Lr) \qquad (10 \cdot 83)$$

Just like the learning identification method, BOP has the characteristic feature that even if computing starts from any initial value, the solution space can be reached in a single round of correlation. It is well known that the necessary number of rounds of multiplication needed in computing Equation (10.83) is proportional to r^2N. Because coefficient correction is performed for every r samples, for each sample, the equivalent number of rounds is on the order of rN. It is believed that by selecting a small value for r, treatment can be effectively performed with current available technology. The specific computing method of BOP and its properties is omitted due to constraints in space. For details, the reader is referred to references [23] and [24]. Also, in

consideration of the influence of the finite length word length operation in practical application, usually, better results are obtained if $X_{rN}^{+}(Lr)e_r(Lr)$ is not calculated too precisely.

10.5.2 BLMS algorithm

As described in Section 10.3, for the LMS, in the steepest descending method, as the estimated values of the various statistical quantities, one has

$$A_{N,N} \to x_N(k)x_N^T(k) \qquad (10 \cdot 84)$$

$$v_N \to x_N(k)z(k) \qquad (10 \cdot 85)$$

and, based on these equations, Equation (10.52) is computed, and, in Equation (10.53), one has

$$j \to k+1 \qquad (10 \cdot 86)$$

Here, the BLMS is obtained using the following relationships:

$$A_{N,N} \to \frac{1}{r}\sum_{k=rL}^{r(L+1)-1} x_N(k)x_N^T(k) \qquad (10 \cdot 87)$$

$$v_N \to \frac{1}{r}\sum_{k=rL}^{r(L+1)-1} x_N(k)z(k) \qquad (10 \cdot 88)$$

Here, r represents the block length. In this case, Equation (10.53) becomes:

$$h_N(j) = h_N(j-r) - \frac{\alpha(j)}{r}\left[\sum_{k=rL}^{r(L+1)-1} x_N(k)x_N^T(k)h_N(j-r) - \sum_{k=rL}^{r(L+1)-1} x_N(k)z(k)\right] \qquad (10 \cdot 89)$$

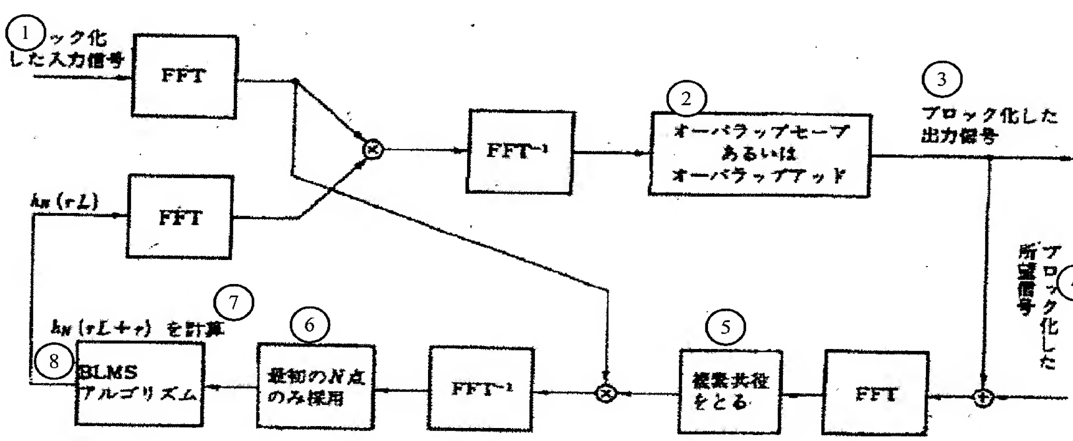


Figure 10.8. Adaptive filter using BLMS.

Key: 1 Block-form input signal
2 Overlap save or overlap add
3 Block-form output signal
4 Block-form desired signal
5 Complex conjugate is taken

6 Only the initial N points are adopted
7 Computing of $h_N(rL + r)$
8 BLMS algorithm

Here, L represents the block No. In addition, assuming that $j = rL + r$ and $\alpha(j) = \alpha$, the algorithm known as BLMS is obtained as follows:

$$h_N(rL+r) = h_N(rL) + \frac{\alpha}{r}\sum_{k=rL}^{r(L+1)-1} x_N(k)\,e(k) \qquad (10 \cdot 90)$$

Here, in $rL \leq k \leq r(L+1)-1$, one has

$$e(k) = d(k) - x_N^T(k)\,h_N(rL) \qquad (10 \cdot 91)$$

Usually, the BLMS is used together with FFT. It is well known that filtering in the frequency region using FFT has the advantage that the number of rounds of multiplication is smaller than that in the time zone [30]. Figure 10.8 is a diagram illustrating the constitution when the BLMS is adopted in the signal processing system based on the frequency region.

In addition, there are also methods that update parameters directly in the frequency region. For details, the reader is referred to references (6), (7), (17), (18), (31). Also, the jump algorithm for increasing the convergence speed of the BLMS will be explained in Section 10.6.

(Hajime Kubota)

10.6 Jump algorithm [27, 32]

The error curved surface of an FIR type adaptive filter is represented by the following secondary form with respect to $\Delta h_n = h_n - w$

$$E[e_n^2] = E[\Delta h_n^T R \Delta h_n] \qquad (10 \cdot 92)$$

$$= E[v_n^T \Lambda v_n] \qquad (10 \cdot 93)$$

$$= \sum_{i=0}^{N-1} \lambda_i E[v_i^2(n)] \qquad (10 \cdot 94)$$

Here, the eigenvalue decomposition of the self-correlated matrix of the input signal is taken as

$$R = V \Lambda V^T, \qquad VV^T = 1$$

$$\Lambda = \mathrm{diag}[\lambda_0, \cdots, \lambda_{N-1}]$$

where $v_n = V^T \Delta h_n$. It is well known that the eigenvalue λ_i is the curvature in the direction of the ith principal axis (eigenvector) of the error curved surface [33]. Consequently, the error curved surface in the principal axis direction where the energy component of the error is large is steep. For such error function, the RLS method as a transformation of the Newton-Raphson method can bring the fastest convergence. On the other hand, when the self-correlation of the input signal is significant, the error energy is concentrated only in some of the eigenvector directions, and the error curved surface is steep in these directions, while it is very mild in the other directions with smaller eigenvalues. When such input correlation matrix approaches degeneracy, the RLS and

other computing methods become unstable. The reason is as follows: the weighting in the flat direction contributes little to the square error, and the overshoot of these weightings cannot be suppressed by minimizing the square error, so overshoot cannot be suppressed. Consequently, it would be ideal to use different updating step sizes in different directions of eigenvalues. On the other hand, when the input signal has high correlation, the gradient vector significantly deviates from the direction toward the optimum solution. Consequently, it is difficult to increase the speed of a gradient computing method that is updated in this direction.

In order to solve the aforementioned problems, a jump algorithm has been proposed. According to this computing method, updating is performed in the gradient direction, while the reciprocal of the eigenvalue of the self correlated matrix of the input signal is used as the step size.

$$h_{n+1}=h_n-\frac{1}{\lambda_i}\nabla_n, \qquad i=0,1,\cdots,N-1 \tag{10・95}$$

Here, ∇_n represents the estimated value of the gradient vector, and $\hat{\lambda}_i$ $(i=0,1,\cdots,N-1)$ is the estimated value of the eigenvalue. In the case of real time treatment, $\hat{\lambda}_i$ is computed using the computing method of DCT, a good approximation of KLT (see Chapter 4), or the like. As time n lapses, the step size repeatedly uses $1/\hat{\lambda}_i$ $(i=0,1,\cdots,N-1)$ in the falling order. A larger one also has a longer application time. $\hat{\lambda}_i$s smaller than a certain lower limit are discarded.

Also, when it is realized as a block computing method, the following form is taken:

$$h_{m+1}=h_m-\frac{1}{N\lambda_i}\sum_{l=0}^{N-1}e(mN+l)\,x(mN+l), \qquad i=0,1,\cdots,N-1 \tag{10・96}$$

here, m represents the block No. In this case, it is possible to use fast computing methods, such as DCT for determining the eigenvalues and sliding FFT to execute a convolution operation. Consequently, the overall computing quantity is similar to that of the block learning identification method.

The jump algorithm is based on the following updating principle. In a FIR type adaptive filter, when coefficient vector h_m is updated using Equation (10.96), the time correlation matrix of the input signal, $R_m=\sum_{l=0}^{N-1}x(mN+l)\,x(mN+l)^T$, and its eigenvalue decomposition are taken as

$$R_m=V_m\Lambda_m V_m^T, \qquad V_m V_m^T=1$$
$$\Lambda_m=\mathrm{diag}[\lambda_0(m),\cdots,\lambda_{N-1}(m)]$$

and one has

$$v_m=V_m\Delta h_m$$
$$v_{m+1}=(I-\mu\Lambda_m)\,v_m \tag{10・97}$$

That is,

$$\begin{bmatrix} v_0(m) \\ v_1(m) \\ \vdots \\ v_{N-1}(m) \end{bmatrix} = \begin{bmatrix} 1-\mu\lambda_0 & & & 0 \\ & 1-\mu\lambda_1 & & \\ & & \ddots & \\ 0 & & & 1-\mu\lambda_{N-1} \end{bmatrix} \begin{bmatrix} v_0(m-1) \\ v_1(m-1) \\ \vdots \\ v_{N-1}(m-1) \end{bmatrix} \quad (10 \cdot 98)$$

When [m] $\mu = 1/\lambda_i$ is used as the step size for updating, the ith component $v_i(m)$ of vector v_m become zero. This is tantamount to v_m going in the gradient direction, and exactly stopping at the crossing point between the extension line of the gradient vector and the orthogonal supplementary space $S_i^{\perp}$ of the ith principal axis. Consequently, the weighting coefficient space becomes partial space $S_i^{\perp}$. Such crossing points are always preset in the gradient direction, and they are the optimum stop positions for updating v_m. Similarly, when the reciprocal of other λ_i is used as the step size, v_m becomes even lower in size, and it enters the orthogonal complementary space $S_{ij}^{\perp}$ between the ith and jth principal axes. Consequently, after N rounds of said updating, v_m reaches the minimum point on the error curved surface.

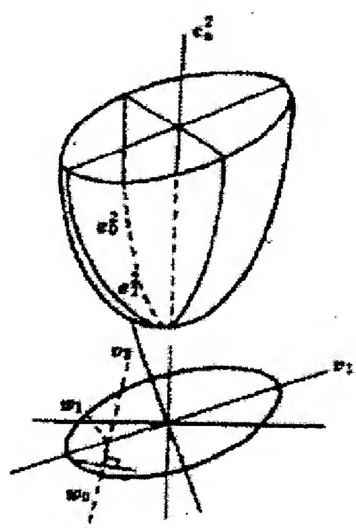


Figure 10.9. Error curved surface and optimum updating position in two-dimensional parameter space.

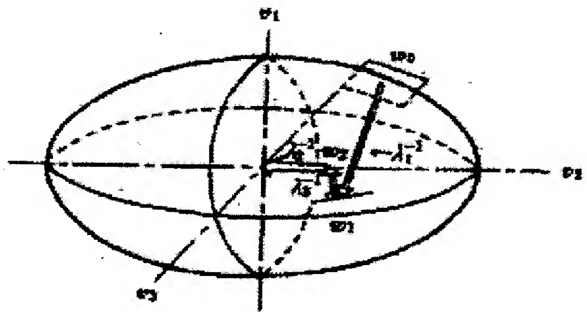


Figure 10.10. Optimum updating route in three-dimensional parameter space.

Figures 10.9 and 10.10 show the optimum stop positions in the gradient direction and the updating process through them. Figure 10.9 shows the updating route in the parameter space when N = 2, and Figure 10.10 shows the case when N = 3.

The following properties of jump algorithms have been found. Because large eigenvalues correspond to the steep direction on the error curved surface, small step sizes as their reciprocals can bring faster convergence. On the contrary, in the direction corresponding to small eigenvalues, the error curved surface is mild, and large step sizes as their reciprocals correspond to a slow convergence mode. Consequently, when a small step size is mainly used in updating, the convergence is faster, and it is possible to obtain an adapting performance with smaller error adjustment.

10.7 IIR type adaptive filter

Usually, when a model is formed for a linear system, a form having a rational transmission function is the most natural. As a matter of fact, when the impulse length is very long, or an unknown system containing damping vibration is taken as an approximation, the use of an IIR type filter having a rational transmission function provides a higher efficiency than that from a scheme using an FIR type filter having a polynomial transmission function. Consequently, research on IIR type adaptive filters has received attention.

In a formulation with an adaptive filter used in system identification, assuming that the input of the unknown system is x_n, the output of the unknown system (the desired signal) is d_n, the observation value of the output signal is z_n, the interference and noise contained in the observation signal are v_n, and the output of the estimation system is y_n. In this case, the transmission function of the unknown system is

$$H(z)=\frac{B(z)}{A(z)}=\frac{b_0+b_1z^{-1}+\cdots+b_Nz^{-N}}{1-a_1z^{-1}-\cdots-a_Nz^{-N}} \quad (10\cdot 99)$$

and the output of this unknown system is

$$y_n = \frac{B(z)}{A(z)} x_n + v_n \qquad (10 \cdot 100)$$

$$= \sum_{i=1}^{N} a_i y_{n-i} + \sum_{j=0}^{M} b_j x_{n-j} + v_n - \sum_{i=1}^{N} a_i v_{n-i} \qquad (10 \cdot 101)$$

(In the following, as shown in the above formulas, the time-zone display method of the transmission function using a polynomial of delay operator z^{-1} will be adopted).

Here, coefficient vector θ, input/output vector φ_n, and the minimum predicted error e_n^* are defined as follows.

$$\theta = \begin{pmatrix} a \\ b \end{pmatrix} = (a_1, \cdots, a_N, b_0, \cdots, b_M)^T \qquad (10 \cdot 102)$$

$$\phi_n = \begin{pmatrix} y_n \\ x_n \end{pmatrix} = (y_{n-1}, \cdots, y_{n-N}, x_n, \cdots, x_{n-M})^T \qquad (10 \cdot 103)$$

$$e_n^* = v_n - \sum_{i=1}^{N} a_i v_{n-i} \qquad (10 \cdot 104)$$

In this case, y_n can be written as

$$y_n = \phi_n^T \theta + e_n^* \qquad (10 \cdot 105)$$

An unknown system in this form is called a linear circuit model [68].

10.7.1 Series/parallel structure

A series/parallel structure is also called the equation error method. According to this method, the observation value of the input signal and the past value of the output are used, and an adaptive estimation is formulated by predicting the current value of the output. Here, the estimated value of coefficient vector θ of the transmission function obtained at time n is taken as θ_n, and, with the square average value of the predicted error (or equation error) minimized:

$$e_n = y_n - \phi_n^T \theta_{n-1} \qquad (10 \cdot 106)$$

$$= A_n(z) y_n - B_n(z) x_n \qquad (10 \cdot 107)$$

the coefficient θ_n at time n of the adaptive filter is updated with the constitution shown in Figure 10.11.

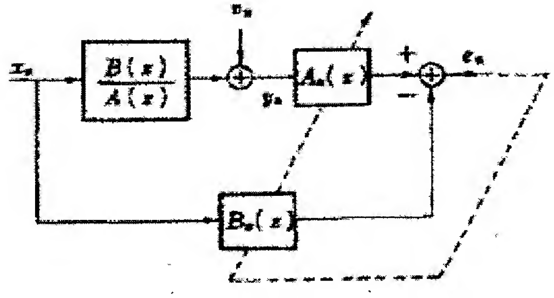


Figure 10.11. Series/parallel type IIR adaptive filter.

In the aforementioned equation, the structure does not contain feedback of the signal to the estimation system itself, and it is not a genuine IIR type. Consequently, the square error becomes a convex function of estimation coefficient θ_n, and the estimation computing method is also similar to that of an FIR type adaptive filter, with the gradient computing method and least square method adopted for it.

In a gradient or LMS series/parallel adaptive filter, the estimation coefficient θ_n is adaptively updated as follows.

$$\theta_n = \theta_{n-1} + \mu e_n \phi_n \qquad (10 \cdot 108)$$

Also, a system has been proposed in which the coefficient vectors of the numerator and the denominator polynomials are updated individually.

$$a_n = a_{n-1} + \mu_a e_n y_n \qquad (10 \cdot 109)$$

$$b_n = b_{n-1} + \mu_b e_n x_n \qquad (10 \cdot 110)$$

The conditions for stability of the various step sizes have been studied [45].

When the RLS method is adopted in an IIR type adaptive filter, by replacing coefficient vector h_n with θ_n and replacing the input vector x_n with input/output vector φ_n, the algorithm described in Section 10.2 can be adopted [68], [70].

For estimation of the aforementioned obtained transmission function, when noise can be ignored and the persistently excitation conditions are also met, it is possible to guarantee approach to the true transmission function or the optimum estimated value most appropriate for its approximation [[illegible]]. Because of major advantages such as guarantee of convergence and fast convergence, the series/parallel type IIR computing method is now widely adopted in automatic equalizers, echo cancellers, etc. [40], [41], [70].

However, in application of an IIR type adaptive filter, in some cases, not only may simple estimation of the transmission function be obtained, but also realization of an approximate system of the unknown system by means of an estimated transmission function. In such case, stability of the estimated transmission function should be guaranteed. This requires that the roots of the denominator polynomial be within a unit circle. Usually, no such guarantee for the

estimated transmission function obtained using the aforementioned method can be made. As a method for guaranteeing that the roots of any denominator polynomial $A_n(z)$ be within a unit circle, there is the Schur-Cohn method. However, in this method, computations on the order of $O(N^2)$ $(N=\deg[A_n(z)])$ are required. The computing quantity is much larger than the computing quantity order of magnitude for estimation. Consequently, it usually becomes the bottleneck for real time processing, so methods to increase the speed of the aforementioned Schur-Cohn method have been investigated [illegible].

10.7.2 Extension of series/parallel computing method and various identification methods

A disadvantage of the aforementioned series/parallel computing method is that a deviation (bias) takes place in the estimated value of parameters when interference noise v_n exists. In order to prevent the aforementioned bias, the following schemes have been proposed as system identification methods for determining an unbiased estimation of parameters: extended least square method (ELS), generalized least square method (GLS), recursive maximum likelihood estimation (RML), and instrumental variable method (IV) (see Chapter 9) [37], [68]. Among the aforementioned methods, for ELS, GLS and RML, in order to display the properties of the interference noise, a noise generation model is assumed for estimation. These methods and IV are included in more general methods for estimating linear recursive models, that is, the recursive prediction error method (RPEM) and pseudo linear regression (PLR) [65].

Here, as the unknown system, the following system is taken in a general linear regression model.

$$A(z)y_n=\frac{B(z)}{F(z)}x_n+\frac{D(z)}{C(z)}v_n. \qquad (10\cdot 111)$$

Here, v_n is taken as white color. For such unknown system, if one defines as follows:

$$\bar{w}_n=\frac{B_n(z)}{F_n(z)}x_n \qquad \bar{v}_n=A_n(z)y_n-\bar{w}_n$$

$$\bar{\varepsilon}_n=\frac{D_n(z)}{C_n(z)}\bar{v}_n \qquad (10\cdot 112)$$

$$\theta=(a^T, b^T, c^T, d^T, f^T)^T \qquad (10\cdot 113)$$

$$\phi_n=(-y_n^T, x_n^T, -\bar{w}_n^T, \bar{\varepsilon}_n^T, -\bar{v}_n^T)^T \qquad (10\cdot 114)$$

$$\psi_n=\left(-\frac{D_n(z)}{C_n(z)}y_n^T, \frac{D_n(z)}{C_n(z)F_n(z)}x_n^T, -\frac{D_n(z)}{C_n(z)F_n(z)}\bar{w}_n^T, \frac{1}{C_n(z)}\bar{\varepsilon}_n^T, -\frac{1}{C_n(z)}\bar{v}_n^T\right)^T \qquad (10\cdot 115)$$

output prediction $\bar{y}_n$ and its prediction error using estimated value θ_n of coefficient vector θ become the following:

$$\hat{y}_n = \theta_n^T \phi_n \qquad (10 \cdot 116)$$
$$\varepsilon_n = y_n - \hat{y}_n \qquad (10 \cdot 117)$$

For such model, the basic algorithm of the recursive prediction error method is as follows:

$$\theta_{n+1} = \theta_n + \gamma R_n^{-1} \varepsilon_n \psi_n \qquad (10 \cdot 118)$$
$$R_{n+1} = R_n + \gamma(\psi_n \psi_n^T - R_n) \qquad (10 \cdot 119)$$

Also, the PRL method becomes

$$\theta_{n+1} = \theta_n + \gamma R_n^{-1} \varepsilon_n \phi_n \qquad (10 \cdot 120)$$
$$R_{n+1} = R_n + \gamma(\phi_n \phi_n^T - R_n) \qquad (10 \cdot 121)$$

In Equation (10.111), when $A(z) = F(z) = C(z) = D(z) = 1$, the aforementioned computing method becomes the RLS method. When $F(z) = D(z) = 1$, the RPEM method becomes the maximum likelihood estimation method RML, and the PLR method becomes the ELS method. Also, the GLS method corresponds to the case of $F(z) = C(z) = 1$, and the HARF and SHARF method to be described in the next section corresponds to the case of $A(z) = C(z) = D(z) = 1$.

In order to estimate the parameters of the aforementioned complicated models, it is necessary to include a closed feedback loop in the estimation system, and to solve a general nonlinear optimization problem. In the aforementioned method, the local optimum solution is determined by means of the probability Newton-Raphson method. Its algorithm is basically a combination of the recursive least square method and the boot-strap method. Consequently, its convergence value also depends on the initial value. In the analysis of convergence of the aforementioned algorithm, the ordinary differential equation (ODE) method, the Liapnov function method, etc. are adopted.

In addition, the instrumental variable method that actively uses the property that noise is not correlated to the input signal can be used. Because an error curved surface is a secondary function of the parameter, one may simply solve a normal equation that has been extended in estimation, and its solution can be uniquely determined. This is a major advantage over the aforementioned methods. This computing method has the same form as that of the RLS method when the inverted matrix rama [transliteration] is used to obtain a recursive solution. This system is now adopted in several fields of adaptive signal processing [58], and the gradient instrumental variable method is also adopted in echo canceling [42], [45].

10.7.3 Parallel structure

The parallel structure is also called the output error method (OEM). It differs from the series/parallel structure in that an IIR type estimation system is formed, and the estimation computing method is derived by minimizing the error between its output $\hat{y}_n$ and the output y_n of the unknown system:

$$\begin{aligned}\varepsilon_n &= y_n - \hat{y}_n \\ &= \frac{B(z)}{A(z)} x_n - \frac{B_n(z)}{A_n(z)} x_n + v_n \end{aligned} \qquad (10 \cdot 122)$$

Figure 10.12 is a diagram illustrating its constitution.

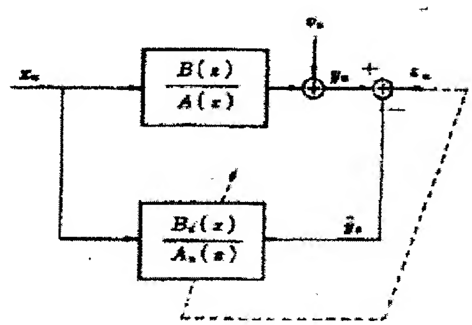


Figure 10.12. Parallel-structure IIR type adaptive filter

[1] Updating computing method by means of gradient method

When the gradient method is used to minimize the square of the output error with respect to θ_n, the parameters are updated as follows.

$$\theta_{n+1} = \theta_n + 2\mu e_n \frac{\partial y_n}{\partial \theta(n)}$$

where, the components of the gradient vector are defined as follows:

$$\alpha_p(n) = \frac{\partial y_n}{\partial a_p(n)}, \qquad p = 1, 2, \cdots, N \qquad (10 \cdot 123)$$

$$\beta_q(n) = \frac{\partial y_n}{\partial b_q(n)}, \qquad q = 0, 1, \cdots, M \qquad (10 \cdot 124)$$

Usually, if the adaptive updating is sufficiently smooth, the following approximations can be adopted:

$$\frac{\partial y_{n-i}}{\partial a_p(n)} \cong \alpha_p(n-i) \qquad (10 \cdot 125)$$

$$\frac{\partial y_{n-i}}{\partial b_q(n)} \cong \beta_q(n-i) \qquad (10 \cdot 126)$$

and the following recursive equations of the gradient vector can be obtained:

$$\alpha_p(n) = y_{n-p} + \sum_{i=1}^{M} a_i(n)\alpha_p(n-i) \qquad (10 \cdot 127)$$

$$\beta_q(n) = x_{n-q} + \sum_{i=1}^{M} a_i(n)\beta_q(n-i) \qquad (10 \cdot 128)$$

However, when the denominator polynomial is $A_n(z^{-1})$, the time-varying filter is contained in the adaptive filter itself and in computing the gradient. Consequently, if it is unstable, the adaptive filter also becomes divergent. Consequently, in order to guarantee stability, if the roots of $A_n(z^{-1})$ obtained at time n are over the unit circle, the new estimated

value is discarded, or the roots are pulled back into the unit circle followed by updating with the methods proposed in [38], [39].

On the other hand, because it is easy to judge the stability of a secondary IIR filter, a scheme to include it in the structure has been proposed. For example, in a scheme proposed in [39], a the secondary parallel structure is prepared by parallel connection of secondary consecutive structures prepared by sequentially connecting secondary IIR filters (biquads).

In addition, a structure using a secondary IIR filter having a fixed pole for guaranteeing stability [44] has also been proposed. The positions of the fixed poles are predetermined according to knowledge obtained in experiments. In addition, a structure using a total polar type lattice filter has been proposed [39].

[2] Output error method by means of hyper-stable theory

The gradient method has the disadvantage that the convergence speed is low. In order to increase the speed of convergence, adoption of the Newton-Raphson method has been considered. However, because the error curved surface of the estimation system containing feedback is a complicated nonlinear function, it cannot be approximated as a secondary function. Consequently, local convergence cannot be guaranteed when the RLS method and other methods are adopted in starting from any initial point.

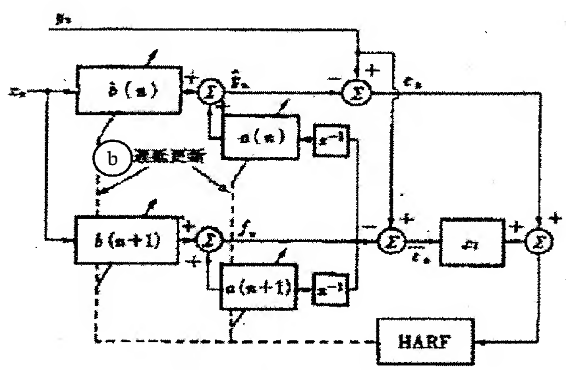


Figure 10.13. Structure of HARF computing method (courtesy Larimore A., et al., Copyright ©1980 IEEE).

Key: b Delay updating

In order to solve the aforementioned problem, a hyperstable adaptive recursive filter (HARF) and a simplified HARF (SHARF) have been proposed as adaptive updating computing methods that guarantee the local convergence property (or large-area stability) by using the

hyperstability theory known as the general stability theory in a nonlinear time-varying system [47], [48]. Figure 10.13 shows the constitution of this scheme.

For the estimation system, the output $\hat{y}_n$ and the instrumental output f_n are as follows:

$$\hat{y}_n = \sum_{i=1}^{N} a_i(n) f_{n-i} + \sum_{j=0}^{M} b_j(n) x_{n-j} \qquad (10 \cdot 129)$$

$$f_n = \sum_{i=1}^{N} a_i(n+1) f_{n-i} + \sum_{j=0}^{M} b_j(n+1) x_{n-j} \qquad (10 \cdot 130)$$

Also, the output error and the smoothed error are defined as follows:

$$\varepsilon_n = y_n - \hat{y}_n \qquad (10 \cdot 131)$$

$$\bar{\varepsilon}_n = \sum_{i=1}^{L} c_i [y_{n-i-1} - f_{n-i-1}] \qquad (10 \cdot 132)$$

Here, the following filter is called the smoothing filter:

$$C(z) = 1 + \sum_{i=1}^{L} c_i z^{-1} \qquad (10 \cdot 133)$$

According to the HARF computing method, the coefficients are updated as follows:

$$a_i(n) = a_i(n-1) + \frac{\mu_i}{q_n} f_{n-i-1} [\varepsilon_n + \bar{\varepsilon}_n],$$
$$1 \le i \le N \qquad (10 \cdot 134)$$

$$b_j(n) = b_j(n-1) + \frac{\rho_j}{q_n} x_{n-j-1} [\varepsilon_n + \bar{\varepsilon}_n],$$
$$0 \le j \le M \qquad (10 \cdot 135)$$

Here, q_n is taken as the normalization factor, and it is obtained as follows:

$$q_n = 1 + \sum_{i=1}^{N} \mu_i f_{n-i-1}^2 + \sum_{j=0}^{M} \rho_j x_{n-j-1}^2 \qquad (10 \cdot 136)$$

When the smoothing filter C(z) meets the following strict positive realness (SPR) and an appropriate initial value of the coefficient is selected, convergence of the HARF computing method can be guaranteed. That is, when $G(z) = C(z)/A(z)$, the following relationship is established:

$$\mathrm{Re}[G(e^{j\theta})] > 0, \qquad \forall \theta \in [0, 2\pi) \qquad (10 \cdot 137)$$

When the initial value is also appropriate, the error quantity

$$y_n - f_n + \sum_{i=1}^{L} [y_{n-i-1} - f_{n-i-1}] \qquad (10 \cdot 138)$$

converges to 0, and one has $\hat{y}_n = f_n = y_n$. Consequently, in practice, the following relationship is established:

$$C(z) = A_n(z) \qquad (10 \cdot 139)$$

10.8 Adaptive lattice filter [50]-[52]

Itakura and Saito have shown that the adaptive lattice filter (Figure 10.14) can realize the Levinson-Durbin computing method as a linear prediction computing method (see Chapter 8) [66]. This structure has the advantage that the sensitivity of the filter coefficient is low, and the

numeric stability is also high. Also, it is simple to guarantee the safety of the total polar structure. Due to these and other advantages, it is now in wide use in various fields of adaptive signal processing [48].

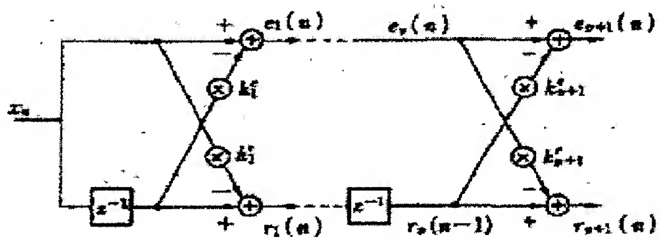


Figure 10.14. Adaptive lattice filter.

10.8.1 Adaptive linear prediction and lattice type filter

First, the pth-order linear prediction problem is formulated. Linear prediction of the current value using p past values of signal x_n is as follows:

$$\hat{x}_{p,n} = -a_1^p(n)x_{n-1} - \cdots - a_p^p(n)x_{n-p} \quad (10 \cdot 140)$$

This prediction is called forward prediction. On the other hand, the prediction of x_{n-p} using $x_n, \ldots, x_{n-p}$

$$\hat{x}_{p,n-p} = -b_1^p(n)x_{n-p+1} - \cdots - b_p^p(n)x_n \quad (10 \cdot 141)$$

is called backward prediction. Here, signal vector x, the forward prediction vector and the backward prediction vector are defined as follows:

$$x_{n,n-p} = (x_n, x_{n-1}, \cdots, x_{n-p})^T \quad (10 \cdot 142)$$
$$A_p(n) = (1, a_1^p(n), \cdots, a_p^p(n))^T \quad (10 \cdot 143)$$
$$B_p(n) = (b_p^p(n), \cdots, b_1^p(n), 1)^T \quad (10 \cdot 144)$$

and, the forward and backward prediction errors are as follows:

$$e_p(n) = x_n - \hat{x}_{p,n} = A_p^T(n)x_{n,n-p} \quad (10 \cdot 145)$$
$$r_p(n) = x_{n-p} - \hat{x}_{p,n-p} = B_p^T(n)x_{n,n-p} \quad (10 \cdot 146)$$

For the lattice structure, the forward prediction error $e_p(n)$ and the backward prediction error $r_p(n)$ are defined as the following order-updated form:

$$e_{p+1}(n) = e_p(n) - k_{p+1}^f r_p(n-1) \quad (10 \cdot 147)$$
$$r_{p+1}(n) = r_p(n-1) - k_{p+1}^r e_p(n) \quad (10 \cdot 148)$$

where k_p^f, k_p^r are reflection coefficients.

10.8.2 Adaptive realization of Levinson computing method

As methods for realizing an adaptive lattice filter, the block computing method and recursive computing method exist. For computing the reflection coefficient, a few methods have been proposed, that is, a method in which the statistical average in the Levinson-Darwin

[transliteration] method is replaced with the time average, and a method in which update is performed using a gradient method to obtain a minimum prediction error for each section.

First, the following computing method can be adopted to compute a reflection coefficient having the minimum sum of squares in the block of forward and backward prediction errors [68].

$$k_{p+1}^{r}=\frac{\sum_{j} r_p(j-1)e_p(j)}{\sum_{j} r_p^2(j-1)} \qquad (10 \cdot 149)$$

$$k_{p+1}^{e}=\frac{\sum_{j} r_p(j-1)e_p(j)}{\sum_{j} e_p^2(j-1)} \qquad (10 \cdot 150)$$

Itakura and Saito use the geometric average of the aforementioned two coefficients in computing a reflection coefficient [69]:

$$k_{p+1}=\frac{\sum_{j} r_p(j-1)e_p(j)}{\sqrt{\sum_{j} r_p^2(j-1)\sum_{j} e_p^2(j-1)}} \qquad (10 \cdot 151)$$

k_p computed in this way is called the PARCOR coefficient, and, because its absolute value is always smaller than one, the minimum phase property of the filter transmission function is maintained. In addition, in order to maintain the minimum phase property, another method uses the minimum values of k_p^r, k_p^e.

The reflection coefficient corresponding to the minimum sum of squares in the block of the forward prediction error and backward prediction error is as follows:

$$k_{p+1}=\frac{2\sum_{j} r_p(j-1)e_p(j)}{\sum_{j} r_p^2(j-1)+\sum_{j} e_p^2(j-1)} \qquad (10 \cdot 152)$$

and this becomes the harmonic average of k_p^r, k_p^e [67].

On the other hand, a gradient recursive computing method has been proposed as a scheme for real time updating of the reflection coefficient in (10.149) and (10.150) [69]:

$$k_{p+1}^{r}(n+1)=k_{p+1}^{r}(n)+\mu r_p(n-1)e_{p+1}(n) \qquad (10 \cdot 153)$$

$$k_{p+1}^{e}(n+1)=k_{p+1}^{e}(n)+\mu r_{p+1}(n)e_p(n) \qquad (10 \cdot 154)$$

or, even simpler forms may be used:

$$k_{p+1}(n)=k_{p+1}^{r}(n)=k_{p+1}^{e}(n) \qquad (10 \cdot 155)$$

$$k_{p+1}(n+1)=k_{p+1}(n)+\mu[r_p(n-1)e_{p+1}(n)+r_{p+1}(n)e_p(n)] \qquad (10 \cdot 156)$$

To normalize them, in Equations (10.153) and (10.154), μ is replaced by $\beta/R_p^r(n)$, $\beta/R_p^e(n)$ to become:

$$R_p^r(n+1)=\lambda R_p^r(n)+r_p^2(n) \qquad (10 \cdot 157)$$

$$R_p^e(n+1)=\lambda R_p^e(n)+e_p^2(n+1) \qquad (10 \cdot 158)$$

Also, in Equation (10.156), another scheme uses $\beta/R_p(n)$ instead of μ:

$$k_{p+1}(n+1) = k_{p+1}(n) + \frac{\beta}{R_p(n)}[r_p(n-1)e_{p+1}(n) + r_{p+1}(n)e_p(n)] \quad (10 \cdot 159)$$

$$R_p(n+1) = \lambda R_p(n) + [r_p^2(n) + e_p^2(n+1)]. \quad (10 \cdot 160)$$

In addition, the recursive updating method of reflection coefficient (10.152) is as follows:

$$\Delta_{p+1}(n+1) = \lambda\Delta_{p+1}(n) + r_p(n-1)e_p(n) \quad (10 \cdot 161)$$

$$k_{p+1}(n+1) = \frac{\Delta_{p+1}(n+1)}{R_p(n+1)} \quad (10 \cdot 162)$$

and $R_p(n)$ is updated as shown in Equation (10.160) [49].

10.8.3 Least square lattice type filter

For least square estimation, it is necessary to solve a normal equation. In this case, the operation for determining the inverted matrix of the co-dispersion matrix of the signal plays a central role. The Gaussian elimination method and other methods require a computing quantity of $O(N^3)$. In consideration of this problem, Levinson has proposed a fast computing method for updating in the order of $O(N^2)$ that uses the property that the co-dispersion matrix in the weak steady state process is a Toeplitz. This is realized by means of lattice type filters (least square lattice filters (LSL)). In addition, in the adaptive processing, the shift-low-rank property of the sample co-dispersion matrix is exploited by Morf in presenting a fast computing method in the order of O(N). This method has since become the basis of the fast least square computing method. The transversal type fast RLS computing method now available is numerically unstable. On the other hand, an LSL computing method based on the same scheme yet having stable time updating and with the same order of computing quantity as that of the gradient method [55] has been proposed.

First, assuming that $A_p(n)$, $B_p(n)$ is the optimum prediction vector for realizing the least square estimation, the following normal equations are established.

$$R_p(n)[A_p(n), B_p(n)] = \begin{bmatrix} \sigma_p^a(n) & 0 \\ \vdots & \vdots \\ 0 & \sigma_p^r(n) \end{bmatrix} \quad (10 \cdot 163)$$

$$R_p(n) = [x_{0,-p}, \cdots, x_{n,n-p}] \begin{bmatrix} x_{0,n-p}^T \\ \vdots \\ x_{n,n-p}^T \end{bmatrix} \quad (10 \cdot 164)$$

Here, $\sigma_p^a(n)$, $\sigma_p^r(n)$ represent the least square errors of the pth-order forward and backward prediction errors.

In the following, an explanation will be given regarding a method in determining the solution of the (p + 1)th-order linear prediction at the same time (order updating) that the solution of the pth-order linear prediction at time n is known, and a method in determining the pth-order prediction at time n + 1 (time updating) when the pth-order prediction at time n is known.

(1) Order updating

The order updating of prediction vectors $A_p(n), B_p(n)$ is performed as follows.

$$A_{p+1}(n)=\begin{bmatrix}A_p(n)\\0\end{bmatrix}-k^r_{p+1}(n)\begin{bmatrix}0\\B_p(n-1)\end{bmatrix} \tag{10・155}$$

$$B_{p+1}(n)=\begin{bmatrix}0\\B_p(n-1)\end{bmatrix}-k^e_{p+1}(n)\begin{bmatrix}A_p(n)\\0\end{bmatrix} \tag{10・166}$$

where,

$$k^r_{p+1}=\frac{\Delta_{p+1}(n)}{\sigma_p^r(n)} \tag{10・167}$$

$$k^e_{p+1}=\frac{\Delta_{p+1}(n)}{\sigma_p^e(n)} \tag{10・168}$$

$\Delta_p(n)$ is the time mutual correlation function of the forward and backward prediction errors. Here, one has

$$\sigma^e_{p+1}(n)=\sigma_p^e(n)-k^r_{p+1}(n)\Delta_{p+1}(n) \tag{10・169}$$

$$\sigma^r_{p+1}(n)=\sigma_p^r(n)-k^e_{p+1}(n)\Delta_{p+1}(n) \tag{10・170}$$

Consequently, the lattice computing method for order updating of the forward and backward prediction errors is defined as follows:

$$e_{p+1}(n)=e_p(n)-k^r_{p+1}(n)r_p(n-1) \tag{10・171}$$

$$r_{p+1}(n)=r_p(n-1)-k^e_{p+1}(n)e_p(n) \tag{10・172}$$

(2) Time updating

First, the likelihood variable $\gamma_p(n)$ is defined as follows:

$$\gamma_p(n)=x^T_{n,n-t}R_p^{-1}(n)x_{n,n-t} \tag{10・173}$$

It has the order updated as follows:

$$\gamma_{p+1}(n)=\gamma_p(n)+\frac{r^2_{p+1}(n)}{\sigma^r_{p+1}(n)} \tag{10・174}$$

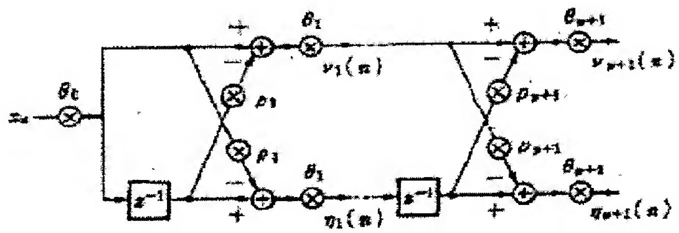
Each time a new signal vector $x_{p,n+1}$ is input, the various variables of the aforementioned lattice computing method are time updated as follows:

$$\Delta_{p+1}(n+1)=\Delta_{p+1}(n)+\frac{e_p(n+1)\,r_p(n)}{1-\gamma_{p-1}(n)} \qquad (10\cdot 175)$$

$$\sigma_p^e(n+1)=\sigma_p^e(n)+\frac{e_p^2(n+1)}{1-\gamma_{p-1}(n)} \qquad (10\cdot 176)$$

$$\sigma_p^r(n+1)=\sigma_p^r(n)+\frac{r_p^2(n+1)}{1-\gamma_{p-1}(n)} \qquad (10\cdot 177)$$

As can be seen from comparison between the gradient lattice computing method and the LSL, although both computing methods compute a mutual correlation function of the forward and backward prediction errors, they are different from each other as follows: while in the gradient method, the weighting of the new signal at various times is uniform, in the LSL method, weighting coefficient $1/(1-\gamma_p(n))$ containing $\gamma_p(n)$ is used. Because one has $0\leq\gamma_p(n)\leq 1$, for the self correlation information of the new signal vector, the more the new component as compared with the co-dispersion matrix of the signals that have been accumulated, the nearer $\gamma_p(n)$ is to unity. Consequently, when a signal having new information is input, its weight becomes very large, so fast convergence is possible.



$$\theta_p=\frac{1}{\sqrt{(1-\nu_p^2(n))(1-\eta_p^2(n-1))}}$$

Figure 10.15. Normalized LSL adaptive filter

(3) Normalized LSL computing method

When the aforementioned LSL computing method is normalized, it is clear that computing of the square root is introduced. As a result, a concise computing method can be obtained for updating the three parameters (normalized reflection coefficient $\rho_p(n)$, normalized forward and backward prediction errors $\nu_p(n),\ \eta_p(n)$)) (Figure 10.15).

$$\rho_{p+1}(n)=\sqrt{1-\nu_p^2(n)}\sqrt{1-\eta_p^2(n-1)}\,\rho_p(n)+\nu_p(n)\eta_p(n) \tag{10・178}$$

$$\nu_{p+1}(n)=\frac{\nu_p(n)-\rho_{p+1}(n)\eta_p(n-1)}{\sqrt{1-\nu_p^2(n)}\sqrt{1-\eta_p^2(n-1)}} \tag{10・179}$$

$$\eta_{p+1}(n)=\frac{\eta_p(n-1)-\rho_{p+1}(n)\nu_p(n)}{\sqrt{1-\nu_p^2(n)}\sqrt{1-\eta_p^2(n-1)}} \tag{10・180}$$

(4) Joint estimation by LSL

The lattice adaptive filter is a computing method developed for linear prediction of a signal. However, for the problem of processing of an adaptive signal that can be formulated as FIR type system identification, the past value of input signal x_n is used to predict the desired signal y_n. That is, joint estimation is necessary.

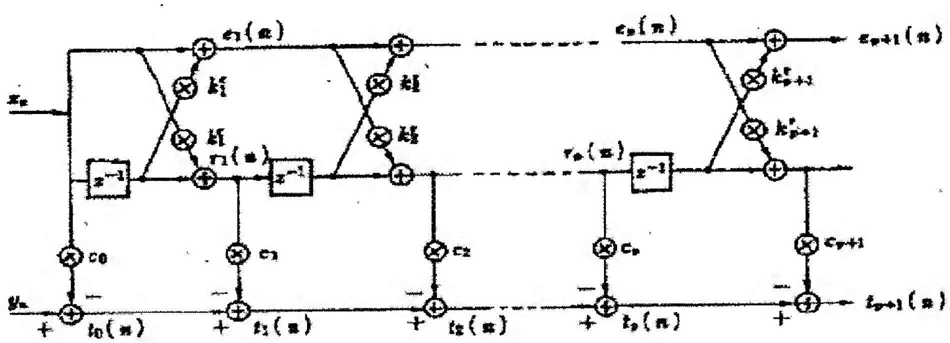


Figure 10.16. Joint estimation using lattice type ADF.

In this case, when the aforementioned LSL computing method is adopted, the least square joint estimation method can be realized easily. Its constitution is shown in Figure 10.16. More specifically, instead of using the past value of input signal x_n, the backward prediction errors $\{r_p(n)\}$ of the orthogonal LSL are used, and y_x is predicted by means of a linear combination of them. Assuming that the prediction error is as follows:

$$t_p(n)=y_n-\sum_{p=0}^{P} c_p(n)\, r_p(n) \tag{10・181}$$

weighting coefficient cp(n) is updated as follows.

$$d_p(n)=d_p(n-1)+\frac{t_{p-1}(n)\, r_p(n)}{1-\gamma_{p-1}(n)} \tag{10・182}$$

$$c_p(n)=\frac{d_p(n)}{\sigma_p^r(n)} \tag{10・183}$$

10.9 Constitutions of other adaptive filters

10.9.1 Constitution with whitening of the input signal

In order to overcome the problem of the gradient method in that it has a low convergence speed with respect to a colored input signal, a method has been proposed in which the input signal is first converted to white color before input to the adaptive filter. As a scheme for whitening, the Gram-Schmidt orthogonal method is often adopted. Also, there is a scheme in which the adaptive lattice filter is used to output the weighted linear coupling of the orthogonal backward prediction error $\{r_p(n)\}$ [57]. Here, the weighting coefficient is updated using the conventional LMS method.

10.9.2 Fast computing method of least square method and systolic array

Recently, in order to further increase the operating efficiency, a fast RLS computing method has been proposed, that is, FTF (fast RLS transversal filter) [58]. However, it has been pointed out that this method is numerically unstable for other than the lattice fast computing method (LSL). Consequently, even LSL should not be used for higher order [62]. On the other hand, research has been performed on a scheme in which plural processors are used, and a parallel processing and pipeline processing are combined to increase the overall throughput in a systolic array method or the like [51].

10.9.3 ARMA lattice filter and identification method by means of embedding

The aforementioned lattice filter has been adopted in predicting in the AR process (see Chapter 8, Section 8.5). Also, a lattice structure that can also predict the ARMA process has been studied (see Chapter 8, Section 8.8) [52]. More specifically, a scheme has been adopted for predicting the 2-channel signal $x_n = (y_n, x_n)^T$ obtained by embedding input/output signals. Especially, according to the ARMA Levinson computing method, prediction of a multi-channel signal can be obtained by using a recirculation lattice filter [[illegible]].

Also, when the problem of identification of an IIR system is converted to the problem of prediction of 2-channel signal x_n, the fast computing method of the multi-channel least square estimation and a lattice filter can be used [59], [60], [70] (in consideration of the causal relationship of the modeling, $(y_{n-r}, x_n)^T$ or another delayed embedding must be used [60], [61]). In addition, by correcting the multi-channel maximum entropy computing method, it is possible to estimate the minimum phase of the denominator polynomial of the transmission function, and a pseudo unknown system with stability guaranteed automatically can be determined [62]. Especially, the constitution of a LATIN (lattice-inverse) structure formed by sequentially connecting a lattice filter and its inverse filter is effective in the applications of IIR-shape echo hauling cancellation, etc. [63], [64].

10.10 Adaptive array

An adaptive filter is often used in processing signals taken as a sequence in time. Also, the signals of an adaptive array are processed as a time-space sequence by means of space characteristics together with time characteristics. For example, two signals having different space characteristics are highly correlated to each other as time sequences, and, when their frequency bands superpose one another, it is hard to separate them with an adaptive filter. However, when an adaptive array using the difference in space between two signal sources is adopted, it is possible to separate said signals. Consequently, the present technology is now widely adopted in various fields, such as radar, sonar, wireless communication, acoustic processing, as well as in geological surveying, astronomical measurements, etc.

As shown in Figure 10.17, a typical structure of a wire-like adaptive array comprises an array of sensors and a conventional complex coefficient adaptive filter connected together (for narrow band signals, usually only a weighting is connected to each sensor. However, in the following, as a general structure, it is assumed that the structure is for wide band signals (wide band array) having FIR type adaptive filters connected to the sensors).

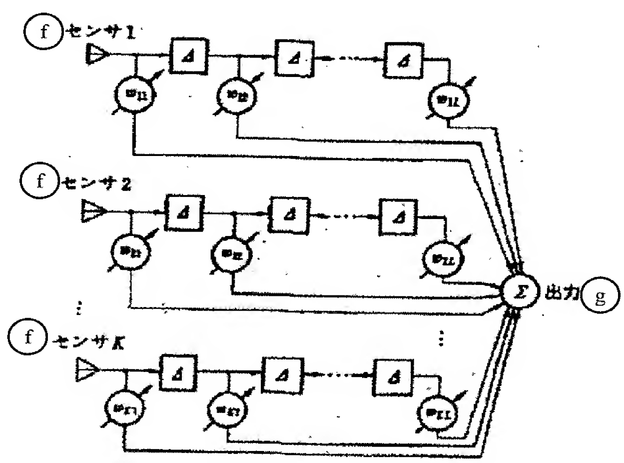


Figure 10.17. Structure of a wide band adaptive array (courtesy Window B. et al., Copyright 1967 IEEE).

Key: f Sensor
g Output

Assuming that the received signal of the sensor is $x_i(n)$, the output signal of the array becomes:

$$y(n)=\sum_{i=1}^{K}\sum_{j=1}^{L} w_{ij}^{*} x_i(n-j+1) \qquad (10 \cdot 184)$$

Here, the coefficient vector and signal vector are defined as follows:

$$w = (w_{11}, \cdots, w_{KL})^T \qquad (10 \cdot 185)$$

$$x_n = (x_1(n), \cdots, x_K(n-L+1))^T \qquad (10 \cdot 186)$$

As a result, one has

$$y(n) = w^H x_n \qquad (10 \cdot 187)$$

Here, w^H is the complex conjugate of w^T.

In the following, the steering vector that represents the intrinsic characteristics of the array will be defined. For a complex plane wave of an incident signal with an arriving angle (the angle with respect to the normal direction of the array) of θ and at a frequency of ω, the steering vector becomes

$$d(\theta, \omega) = (1, e^{j\omega\tau_1(\theta)}, \cdots, e^{j\omega\tau_{KL}(\theta)})^T \qquad (10 \cdot 188)$$

where, $\tau_i(\theta)$ represents the delay in the signal from the first tap to the ith tap.
The steering of the array is represented by the inner product of the coefficient vector and the steering vector.

$$r(\theta, \omega) = w^H d(\theta, \omega) \qquad (10 \cdot 189)$$

Consequently, by appropriate adjustment of the relative direction of coefficient vector ω and the steering vector, it is possible to suppress or extract an input signal in a prescribed direction and at a prescribed frequency.

In the following, an explanation will be given regarding several computing methods wherein the coefficient vector ω is defined adaptively. Just like conventional adaptive computing methods, they can be classified into a block type and recursive type.

10.10.1 Side lobe canceller

Side lobe cancellers each comprise a main canceller and a secondary canceller [71], [72]. The space and frequency characteristics of the principal channel are designed such that the desired signals can be extracted, and an unknown interference signal contained in the output is removed by adaptively adjusting the weighting of the secondary channel. Assuming that the input of the secondary channel is x_c, secondary array coefficient w_a is defined such that the following average electric power of the output y(n) of the entire array reaches a minimum:

$$E\{|y(n)|^2\} = E\{|y_m(n) - w_a^H x_a(n)|^2\} \qquad (10 \cdot 190)$$

Just as in the case of derivation of the FIR type adaptive filter, a determination is possible by solving normal equations:

$$w_a = R_a^{-1} r_{am} \qquad (10 \cdot 191)$$

$$R_a = E(x_a x_a^H), \qquad r_{am} = E(y_m^*(n) x_a) \qquad (10 \cdot 192)$$

As a more specific method for realization, in the case of the block type computing method, the collection average is replaced with the time average in the block. In the case of the recursive computing method, one may adopt the LMS method or RLS method.

If the desired signal is relatively weak, it is possible to improve the SN ratio using this scheme. However, when the desired signal is more intense than the interference signal, the desired signal itself might be eliminated. Usually, the following method is adopted: if a desired signal does not exist, the coefficient of the secondary channel is trained by means of adaptive updating, and, if a desired signal exists, the adaptive updating is stopped.

10.10.2 Method using reference signal [[illegible]]

In practice, when information pertaining to the desired signal $y_d(n)$ is used, it is possible to determine a reference signal $y_r(n)$ similar to the desired signal. In this case, the coefficient of the secondary array of the side lobe canceller can be defined such that the following square error of the output and the reference signal reaches a minimum:

$$E\{|e^2(n)|\} = E\{|y_r(n) - y(n)|^2\} \qquad (10 \cdot 193)$$
$$= E\{|y_r(n) - w^H x(n)|^2\} \qquad (10 \cdot 194)$$

That is,

$$w = R_x^{-1} r_{rx} \qquad (10 \cdot 195)$$
$$R_x = E(x(n) x^H(n)), \qquad r_{rx} = E(y_r^*(n) x(n)) \qquad (10 \cdot 196)$$

When this computing is realized adaptively, just as described in the preceding section, one may adopt the standard LMS method or RLS method.

When the reference signal is generated, one may only have knowledge about the mutual correlation between the desired signal $y_d(n)$ and the input signal. That is, when the following relationship is met:

$$r_{rx} = r_{dx} \qquad (10 \cdot 197)$$

the optimum coefficient can be determined using the aforementioned computing method. Consequently, usually, information about the arrival direction of the desired signal is not required.

10.10.3 Linear constrained minimum variance array [73]

Usually, for an input signal in a prescribed direction and at a prescribed frequency, it is convenient to assign the answer of the array in advance. This gives the constraining condition when the array coefficients are determined. The array coefficients are determined such that the average power of the output reaches a minimum under the constraining condition. For example, for an input signal with arrival angle θ_0 and at frequency ω_0, assuming that the steering vector of

the array is $d(\theta_0, \omega_0) = c$, and the answer of the array is to be set at $w^H c = g$, one may minimize the next evaluation function by means of the Lagrange multiplier method.

$$L_a = w^H R_x w + \lambda\{c^H w - g^*\} \qquad (10 \cdot 198)$$

The obtained array coefficient becomes the following:

$$w = g^* \frac{R_x^{-1} c}{c^H R_x^{-1} c} \qquad (10 \cdot 199)$$

In addition, general constraining conditions, such as assignment of the zero point and fixed gain with respect to plural signals and control of the bandwidth of the assigned gain using a constraining condition with respect to differentiation, etc., are represented by associated linear equations, and are given by the following constraining matrix C and gain vector g.

$$C^H w = g \qquad (10 \cdot 200)$$

Under such constraining condition, the evaluation function

$$L_C = w^H R_x w + \lambda\{C^H w - g\} \qquad (10 \cdot 201)$$

is minimized, and the obtained array coefficient becomes

$$w \doteq R_x^{-1} C [C^H R_x^{-1} C]^{-1} g \qquad (10 \cdot 202)$$

The aforementioned optimum coefficient can be realized by means of a block type adaptive computing method. Also, it is possible to realize the recursive form by means of a gradient method for minimizing L_c. For example, under the single constraining condition $w^H c = g$, updating of the coefficient is performed as follows [75]:

$$w_{n+1} = w_n - \mu P R_x w_n \qquad (10 \cdot 203)$$

$$P = I - \frac{cc^H}{c^H c} \qquad (10 \cdot 204)$$

Here, P is the orthogonal mapping function element with respect to the orthogonal complementary space of c.

Consequently, updating of the coefficient is performed in the orthogonal mapping direction to a region that meets the constraining condition of gradient $R_x w_n$ with respect to w_n of $E(|y(n)|^2)$. Here, R_x is approximated by $x(n)x^H(n)$, and, by using $y(n) = w^H(n)x(n)$, the following updating computing method is obtained [73]:

$$w_{n+1} = w_n - \mu P x(n) y(n) \qquad (10 \cdot 205)$$

Also, with plural constraining conditions, the adaptive computing method may be adopted just as aforementioned.

When the linear constrained minimum variance array (LCMV) is formulated as follows, as a natural extension of the side lobe canceller, a generalized side lobe canceller (GSC) is obtained [74].

First, when the coefficient space is orthogonally decomposed to the value zone space and zero space of restrained matrix C, coefficient vector w becomes the sum of component w_c in the value zone space of C and component -v in the zero space of C.

$$w = w_C - v \qquad (10 \cdot 206)$$

w_c is uniquely defined by the constraint condition:

$$w_C = [CC^H]^{-1} Cg \qquad (10 \cdot 207)$$

Also, assuming that the matrix obtained by taking the base vector of the zero space of C as a column vector is D, it is represented as $v = Dw_D$. Consequently, the topic of estimation becomes a topic for determining w_D. This can be regressed to a constraint-free miniaturization problem. That is, w_D is determined so that the following reaches a minimum:

$$(w_C - Dw_D)^H R_x (w_C - Dw_D) \qquad (10 \cdot 208)$$

The solution obtained using this block computing method is as follows:

$$w_D = (D^H R_x D)^{-1} D^H R_x w_C \qquad (10 \cdot 209)$$

Also, if the recursive computing method is adopted, the following gradient computing method is used:

$$w_{n+1} = w_n - \mu D^H x(n) y^*(n) \qquad (10 \cdot 210)$$

$$y(n) = w_C^H x(n) - w_D^H D^H x(n) \qquad (10 \cdot 211)$$

(J. Chao)

10.11 Cancellation of noise

In the following, an explanation will be given regarding an adaptive echo canceller and an adaptive noise canceller as examples of application of adaptive signal processing. Echo cancellation and noise elimination have the basic idea that instead of estimating the waveform itself, the transmission function (impulse response) of a certain linear system preset in the path of the waveform is estimated. In other words, one should understand that by estimating a parameter of the unknown system, it is possible to eliminate a waveform not in use. Also, in practical application, matching the estimated state of the parameter, the step gain, etc., of the adaptive algorithm are adjusted (see the concept of an intelligent adaptive filter [76], [77]), and it is important to add a function for quick coping with noise, variation in an unknown parameter or an increase in bias.

10.11.1 Adaptive echo canceller (see Book 4, Chapter 5, Section 5.3)

When a long distance telephone line, such as one for international calls, is used, after a few seconds, the voice of the sender will be heard by the sender himself/herself from his/her own receiver, so conversation becomes difficult. The delayed voice of the caller himself/herself is known as echo. The cause for generation of echo is mismatch in impedance of a hybrid circuit set at the connecting portion of a 2-wire line and a 4-wire line. In conventional technology, silence of the counterpart is detected, and a switch known as a voice switch is used to lower the gain of the channel where the voice of the counterpart is sent, so the echo problem is suppressed. However, this system cannot suppress echo (known as double talk) when the counterpart speaks

at the same time as the caller, and switching of the aforementioned voice switch leads to an unnatural sensation. Due to these problems, there is room for further improvement.

With this background, a system has been proposed with the following features: the parameter estimation concept is used for adaptive estimation of the echo path, a pseudo echo is generated and it is subtracted from the original echo so that the echo problem can be suppressed. This system is called an adaptive echo canceller. Figure 10.18 is a diagram schematically illustrating an echo canceller. As shown in Figure 10.18(b), if the transmission function from point A to point B (mainly the impulse response) can be estimated, the pseudo echo can be obtained.

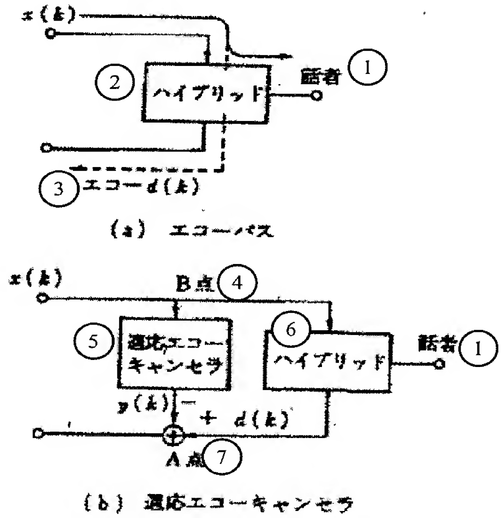


Figure 10.18. Schematic diagram of adaptive echo canceller.

Key:

Key:	(a)	Echo path
	(b)	Adaptive echo canceller
	1	Speaker
	2	Hybrid
	3	Echo
	4	Point B
	5	Adaptive echo canceller
	6	Hybrid
	7	Point A

10.11.2 Adaptive noise canceller

Consider the case when the desired signal is buried in noise. Signal processing technology for minimizing the influence of noise on a desired signal has an extremely wide application range, and it is an old topic but a new research theme. Here, if only the noise can be

retrieved, this topic can be addressed at high efficiency by means of the concept of an adaptive noise canceller. Figure 10.19 is a schematic diagram illustrating an adaptive noise canceller.

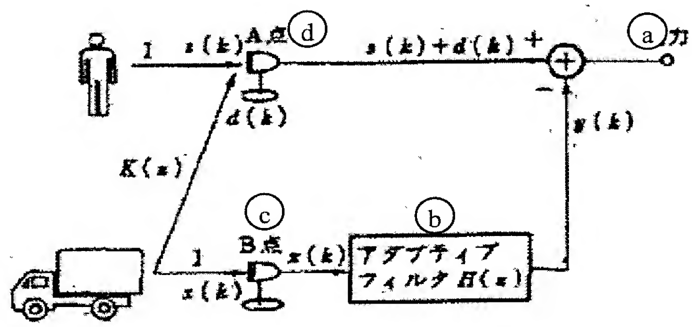


Figure 10.19. Schematic diagram illustrating an adaptive noise canceller.

Key: a Output
b Adaptive filter H(x)
c Point B
d Point A

Here, point A and point B are known as the principal input terminal and reference input terminal, respectively. The principal input terminal is a conventional input terminal, where the waveform as a sum of desired signal s(k) and noise d(k) is input. When the desired signal s(k) is not mixed in the reference input terminal, by subtracting the waveform at the reference input terminal, which has been subjected to appropriate linear treatment, from the waveform at the principal input terminal, it is possible to retrieve only the desired signal. In this example, when transmission function H(z) of the adaptive filter is equal to the path transmission function K(z), the output of the adaptive filter is equal to the noise waveform at the principal input terminal, and it is possible to completely cancel the noise by subtraction.

(Hajime Kubota)

References

(1) H. Itakura and Y. Nishikawa: "About echo cancellation characteristics of echo canceller using learning identification method," Shingakuron (A), J60.A, 11, pp. 1015-1022 (1977).

(2) S. Yamamoto, S. Kishiyama, J. Tamura, K. Chihara, and Y. Sakurai: "Algorithm of echo canceller using fast Karman filter and its characteristics," Shingakuron (B), J64.-B, 4, pp. 318-325 (1981).

(3) H. Itakura and Y. Nishikawa: "Algorithm of echo canceller using Karman filter and its simplification," Shingakuron (A), J62-A, 1, pp. 50-57 (1979-01).

(4) Gersho A.: "Adaptive Equalization of Highly Dispersive Channels for Data Transmission", B. S. T. J., 48, pp. 55-71 (1969)

(5) Widrow B., et al.: "Adaptive Noise Cancelling: Principles and Applications", Proc. IEEE, 63, pp. 1692-1716 (1975)

(6) K. Oishi and H. Kubota: "About simplification of adaptive algorithm of automatic equalizer in frequency region," Shingakuron (A), J70-A, 3, pp. 570-572 (1987-03).

(7) K. Oishi and H. Kubota: "Fast algorithm of automatic equalizer using evaluation quantity in frequency region," Shingakuron (A), J71-A, 3, pp. 867-874 (1988-03).

(8) H. Kubota, T. Furukawa, and H. Itakura: "Algorithm of noise canceller containing pretreatment and evaluation of its performance," Shingakuron (A), J69-A, 5, pp. 584-591 (1986-05).

(9) T. Furukawa, H. Kubota, and H. Matsumoto: "Proposal of signal processing system having high reliability and evaluation of its performance," Shingakuron (A), J73-A, 1. pp. 26-34 (1990-01).

(10) Widrow B. and Hoff M. E.: "Adaptive Switching Circuit", IRE WESCON Conv. Rec., pp. 96-104 (1960)

(11) Nagumo J. and Noda A.: "A Learning Method for System Identification", IEEE Trans. Autom. Control, AC-12, 3, pp. 282-287 (1967)

(12) J. Noda and H. Nagumo: "Learning identification method of system," Keisoku to Seigyo, 7, 9, pp. 597-605 (1968-08).

(13) Kalman R. E.: "A New Approach to Linear Filtering and Prediction Problems", J. Basic Eng. Trans. ASME, 82, pp. 35-45 (1960)

(14) S. Aiyoshi, K. Akitsugi, T. Noka[illegible], and B. Katayama: "System identification," Keisoku Jido Seigyo Gakkai (1981).

(15) H. Sakai: "Signal processing and system identification – about recent algorithms --," Shisutemu to Seigyo, 31, 5, pp. 333-340 (1987-05).

(16) Clark G. A., Mitra S. K. and Parker S. R.: "Block Implementation of Adaptive Digital Filters", IEEE Trans. Circuit & Systems, CAS-28, pp. 548-592, (1981)

(17) Clark G. A., Parker S. R. and Mitra S. K.: "A unified approach to time and frequency-domain realization of FIR adaptive digital filters", IEEE Trans. Acoust., Speech & Signal Process., ASSP-31, 5, pp. 1073-1083 (Oct. 1983)

(18) Walzman T. and Schwartz M.: "Automatic equalization using the discrete frequency domain", IEEE Trans., Inf. Theory, IT-19, 1, pp. 59-68 (Jan. 1973)

(19) Picchi G. and Prati G.: "Self-orthogonalizing adaptive equlization in the discrete frequency domain", IEEE Trans. Commun., COM-32, 4, pp. 371-379 (April 1984)

(20) T. [illegible]moto and S. Maekawa: "Extended learning identification method," Dengakuron (C), 96-C, pp. 227-234 (1975-10).

(21) K. Oseki and S. Umeda: "Adaptive filter algorithm using orthogonal mapping to fine space and its properties," Shingakuron (A), J67-A, 2, pp. 126-132 (1984-02).
(22) T. Furukawa, H. Kubota, and S. Tsujii: "Orthogonal mapping algorithm in block adaptive processing and its properties," Shingakuron (A), J71-A, 12, pp. 2138-2146 (1991-12).
(23) K. Oishi, H. Kubota, and S. Onoda: "Block adaptive algorithm for fast system identification," Shingakuron (A), J71-A, 11, pp. 2018-2026 (1991-11).
(24) T. Furukawa, H. Kubota and S. Tsujii: "Fast block adaptive algorithm and evaluation of its properties," Shingakuron (A), J72-A, 7. pp. 1069-1076 (1989-07).
(25) K. [illegible]mizu and E. Aiyoshi: Suri Keikaku Ho [Mathematical planning method], published by Shokodo, (1984).
(26) T. Nakamizo: "Identification method of linear discrete time system, I. Estimation by means of least square method," Shisutemu to Seigyo, 25, 8, pp. 476-489 (1984-08).
(27) J. Chao, E. Peres [transliteration], and S. Tsujii: "Proposal of jump algorithm in adaptive signal processing," Shingakuron (A), J73-A, 7, pp. 1196-1206 (1990-07).
(28) Ljung L., Morf M. and Falconer D. D.: "Fast Calculation of Gain Matrices for Recursive Estimation Scheme", Int. J. Control, 27, 1, pp. 1-19 (1978)
(29) S. Haykin (translated by M. Takebe): "Introduction to adaptive filters," published by Gendai Kogakusha (1987).
(30) Bowen B. A. and Brown W. R.: VLSI Systems Design for Digital Signal Processing, vol. I, Prentice-Hall (1982)
(31) Dentino M. J., McCool J. and Widrow B.: "Adaptive Filtering in The Frequency Domain", Proc. IEEE, 66, pp. 1658-1659 (1978)
(32) Chao J., Perez H. and Tsujii S.: "A fast adaptive filter algorithm using eigenvalue reciprocals as stepsizes", IEEE Trans. ASSP, 38, 8, pp. 1343-1352 (Aug. 1990)
(33) Widrow B. and Sterns S. D.: Adaptive Signal Processing, Prentice-Hall, Inc. (1985)
(34) Cowan, ed.: Adaptive Filters, Prentice-Hall, Inc. (1985)
(35) Haykin S.: Adaptive Filter theory, Prentice-Hall (1986)
(36) T. Taniogi: Deijitaru shingo kyori no riron 3 [Digital signal processing theory 3], published by Corona Corp. (1986).
(37) T. Nakamizo: "Identification method of linear discrete time system I-IV," Shisutemu to Seigyo, 25, 8, pp. 476-789; 25, 9, pp. 551-563; 25, 10, pp. 609-622; 25, 12, pp. 755-767 (1981); 25, 2, pp. 84-95 (1982).

(38) Johnson R.: "IIR Adaptive Filtering: Current results and open issues", IEEE Trans. IT, 30, 2, pp. 237-250 (March 1984)

(39) Shynk J. J.: "Adaptive IIR Filtering", IEEE ASSP Magazine (April 1989)

(40) Giltin M. M. and Thomson J. S.: "A new structure for adaptive digital echo cancellation", Proc. Nat. Telecomm. Conf., pp. 8. 2. 1-8. 2. 7 (1976)

(41) Ardalan S. H.: "Pole/zero fast Kalman echo cancellation and application to actual measured telephone echo", Proc. IEEE ICC'85, (Chicago, IL), pp. 1482-1486 (1985)

(42) J. Chao, J. Kawabe, S. Tsujii: "New IIR type echo canceller based on auxiliary variable method," Shingaku Giho [Technical Report of IEICE], DSP89-26 (1989).

(43) J. Chao, J. Kawabe, and S. Tsujii: "Studies on IIR type echo canceller GIVE (part 2) – Oblique mapping computing method, convergence property and application in recursive system," Shingaku Giho [Technical Report of IEICE], CAS90-4, DSP90-8 (1990).

(44) [illegible] Uchi[illegible] and M. Hata: "A form of polar control IIR type ADF," Shingakuron, J71-A, 2, pp. 395-403 (1988).

(45) K. Kurozawa, Y. Ido, and S. Tsujii: "Discussion of IIR type learning identification method," Shingakuron (B), J68-b, 11, p. 1229 (1985).

(46) K. Kurozawa: "Fast stability judgment method of discrete time system," Shingakuron (D), J68-d, 11, p. 1967 (1985).

(47) Johnson C. R., Jr.: "A convergence proof for a Hyperstable Adaptive Recursive Filter", IEEE Trans. IT, 25, 6, pp. 745-749 (Nov. 1979)

(48) Larimore M. G. and Treichler J. R. and Johnson C. R., Jr.: "SHARF: An algorithm for adapting IIR digital filters", IEEE Trans. ASSP, 28, 4, pp. 428-440 (Aug. 1980)

(49) Friedlander B.: "Lattice filters for adaptive processing",

(50) H. Sakai: "Signal processing by lattice filter," Shisutemu to Seigyo, 26, 12, pp. 747-755 (1982).

(51) H. Sakai: "Signal processing and system identification – about recent algorithms," 31, 5, pp. 333-340 (1987).

(52) H. Sakai: "Recent topics of lattice filter," Shisutemu to Seigyo, 29, 5, pp. 465-472 (1990).

(53) Sakai H.: "Circular lattice filtering using Pagano's method", IEEE Trans. ASSP, 30, 2, pp. 279-287 (1982)

(54) H. Sakai: "ARMA modeling by circulatory lattice method," Shisutemu to Seigyo Joho Gakkairon, 2, 3, pp. 88-96 (1989).

(55) Lee D. T. L., Morf M. and Friedlander B.: "Recursive least ladder estimation algorithms", IEEE Trans. ASSP, 29, 3, pp. 627-641 (June 1981)
(56) Cioffi J. M. and Kailath T.: "Fast, recursive-least-squares transversal filters for adpative filtering", IEEE Trans. ASSP, 32, pp. 304-337 (1984)
(57) Chang R. W.: "A new equalizer structure for fast start-up digital communications", B. S. T. J., 50, 6, pp. 1969-2014 (July/Aug. 1971)
(58) Friedlander B.: "System identification techniques for adaptive noise cancelling", IEEE Trans. ASSP, 30, pp. 699-708 (1982)
(59) Lee D. T. L., Friedlander B. and Morf M.: "Recursive ladder algorithms for ARMA modeling", IEEE Trans. AC, 27, 8, pp. 753-764 (Aug. 1982)
(60) Kurosawa K., Chao J. and Tsujii S.: "An IIR type of echo canceler using 2-channel lattice filter", Proc. IEEE GLOBECOM'84, pp. 7. 6. 1-7. 6. 6 (1984)
(61) Chao J. and Tsujii S.: "On estimation of noncausal vector AR processes", Int. Joint Tech. Conf. JTCCAS'87, Tokyo (1987)
(62) Chao J. and Tsujii S.: "Exact stable IIR MIMO adaptive identification algorithms", IEEE Trans. CAS, 36, 6, pp. 880-884 (June 1990)
(63) Chao J. and Tsujii S.: "A stable and distortion-free IIR echo and howling cancellor", IEEE Trans. SP, 39, 8, pp. 1812-1821 (Aug. 1991)
(64) J. Chao and S. Tsujii: "Proposal for a controllable 3-terminal pair echo canceller in double talk period," Shingakuron (A), J71-A, 2, pp. 379-386 (1988).
(65) Makhoul J.: "Stable and efficient lattice digital filters: Properties and applications", IEEE Trans. on ASSP, 25, pp. 423-428 (Oct. 1977)
(66) Itakura F. and Saito S.: "Digital filtering techniques for speech analysis and synthesis", Paper 25-C-1, in Proc. 7th Int. Conf. Acoust., pp. 261-264 (1971)
(67) Burg J. P.: "Maximum entropy spectral analysis", Ph. D dissertation, Stanford Univ. (1975)
(68) Ljung L. and Soderstrom T.: Theory and practice of recursive identification, MIT Press (1983)
(69) Griffiths L. J.: "A continuously adaptive filter implemented as a lattice structure", Proc. of ICASSP'77, pp. 683-686 (1977)
(70) Falconer D. D. and Ljung L.: "Application of fast Kalman estimation to adaptive equalization", IEEE Trans.

... is required, and the processing is complicated. For the transmission line code, in order to reduce the influence of near-end crosstalk, it is necessary to attempt to reduce the transmission rate. In the U.S.A., the 2B1Q (2 binary 1 quaternary) code of the 4-value code (transmission rate of 80 kbaud) has been standardized.

For a subscriber line transmission system, without international unification, both the time division direction control transmission system proposed by Japan in the appendix of CCITT recommendations and the echo canceller transmission system proposed by North America and Europe will be described.

[2] Structure and realization

DSU comprises terminal circuits of waveform equalization and timing extraction, etc., and interface circuits with terminals. Among the structure circuits, the functions and operations of an automatic equalizer and echo canceller in which digital signal processing technology is adopted will be explained.

For a subscriber line of metallic cable, a loss occurs that increases in proportion to the square root of the frequency. Consequently, a $\sqrt{f}$ equalizer should be present for compensation. The compensation quantity is defined according to the maximum line loss. For example, for a subscriber line in Japan, with a line loss of 50 dB (160 kHz), it can be included in 99% of the subscriber line distribution that has been set. Also, for a subscriber line, in consideration of the convenience of wiring, when there is a demand for telephone service, a branching line known as a bridge tap is set. However, because the tip of the bridge tap is released, reflection leads to distortion in the waveform. In order to compensate for this, a bridge tap equalizer is needed, and a judgment feedback type equalizer is used. The aforementioned automatic equalizer is needed in both a time division direction control transmission system and an echo canceller transmission system. Usually, it is realized with an analog circuit in the former stage of the A/D converter of the $\sqrt{f}$ equalizer.

In an echo canceller transmission system, in order to reproduce data from a received signal that has been attenuated by 40 dB or more, an echo canceller having an echo canceling property of 60 dB or higher is needed. The echo canceller is usually of the FIR type (about 30 taps). When 2B1Q containing a DC component is used as the transmission line code, in order to suppress the long tail of the echo generated due to influence of the DC cutoff of the transformer of the 2-wire/4-wire conversion hybrid, a combination of an FIR type and an IIR type (primary ~ secondary order) is adopted. In addition, when the echo path has nonlinear characteristics, it is necessary to suppress nonlinear echo. However, it is impossible to suppress nonlinear echo in an FIR type based on conventional linear operation. Consequently, usually, a RAM table type is

used at the same time. According to the RAM table system, the transmission pulse pattern is taken as the address, and the echo level is accumulated so that a pseudo echo is generated. In this system, it is possible to suppress nonlinear echo. However, in order to cut the necessary RAM capacity, a scheme for dividing the RAM is needed [[illegible]].

In order to reduce the size and to lower the power consumption of the DSU, it is necessary to form an LSI for the digital circuit and analog circuit including the automatic equalizer and the echo canceller.

(S. Yamasaki)

5.3 Echo canceller

5.3.1 Generation of echo

One's voice goes through the line on the counterpart side and then returns to the caller himself/herself, and it is heard by the caller as an "echo". This phenomenon is known as caller's echo (hereinafter to be referred to as echo). This echo occurs and hampers conversation in an international call or another call with a significant delay in transmission and in a voice-amplified conversation using a speaker/microphone (hands-free call). Also, when echo is generated at the two ends of a line, a closed loop is formed via the communication network, and, when the loop gain is greater than one, oscillation (hauling) takes place, and communication becomes impossible.

Figure 5.14 is a diagram illustrating the constitution of an analog telephone network circuit. For a subscriber line that connects a subscriber and the local exchange, in consideration of economy, a 2-wire line having the sending signal and the receiving signal superposed on the same pair of wires of the line is adopted. On the other hand, for a long-distance line connecting exchange facilities of different cities, in order to compensate for transmission loss and to use plural lines at a higher efficiency, two pairs of transmission lines are respectively used for the sender side and receiver side to form a 4-wire line system. A 2-wire/4-wire converter that connects said 2-wire line and 4-wire line is a hybrid transformer.

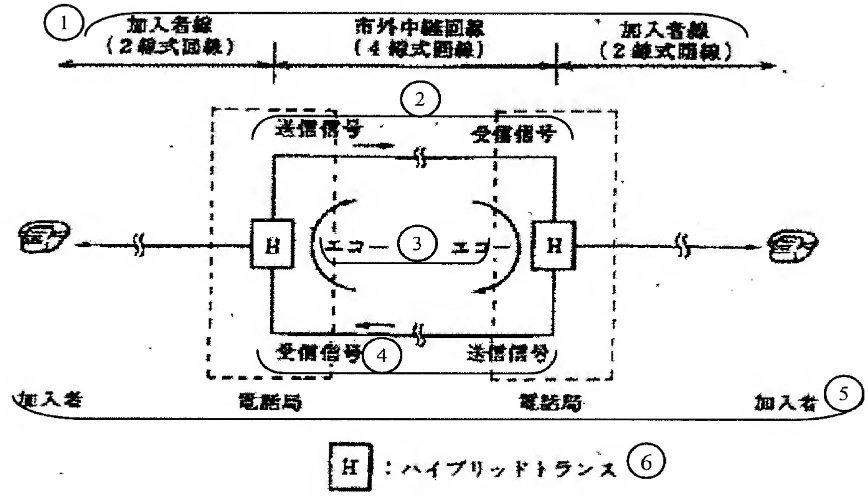


Figure 5.14 Constitution of telephone line.

Key:
1 Subscriber line (2-wire type line)
Local relay line (4-wire line)
Subscriber line (2-wire line)
2 Sending signal
Receiving signal
3 Echo
4 Receiving signal
Sending signal
5 Subscriber
Telephone station
Telephone station
Subscriber
6 Hybrid transformer

The aforementioned hybrid transformer is designed for impedance matching between the 2-wire line and the network for balance, so that the receiving signal on the 4-wire line side does not detour and enter the sending signal on the 4-wire line side. However, different subscribers on the 2-line side have different types of lines and different lengths, so impedance mismatch takes place with the network for balance, and thus a portion of the receiving signal on the 4-wire line side flows into the sending signal on the 4-wire line side, leading to generation of echo.

Echo canceling quantity and echo canceling time

The echo canceling quantity (ERLE) is a quantity indicating the attenuation of the echo quantity, and it is defined as the ratio of the residual signal power ($(e(j))^2$ in Figure 5.18) to the echo signal power ($(y(j))^2$ in Figure 5.18). Also, the echo canceling time is the time until a receiving signal is again input as a sending signal via the echo path. Figure 5.15 is a diagram

illustrating an example of measurement of a one-cycle delay time and the necessary echo canceling quantity [23]. Here, one-cycle delay time refers to the time for the sending signal of the caller to reach the callee and then return to the caller as an echo. As explained above with respect to the definition of echo, a one-cycle delay corresponds to the echo, so the desired echo canceling time is proportional to the overall one-cycle delay time of the entire communication system. For the echo canceling time required for a satellite line and international line, in consideration of the echo canceling in a particular country, it is set at double the longest domestic transmission delay time. For example, Tokyo and Ogasawara Islands are connected via a satellite communication line, and the echo canceling time is 60 ms. Here, the echo canceling quantity is determined to meet the following application. The one-cycle delay time for an international line with a satellite line is about 250 ms and that with a US-Japan marine cable line is about 150 ms. Because the loss quantity for only a conventional domestic transmission system is insufficient, the echo canceling quantity is defined as 30 dB according to CCITT Recommendation G.165. Also, along with the introduction of the domestic relay system of digital optical fiber transmission lines in recent years, due to an increase in the transmission delay and a decrease in the transmission loss, an echo canceller has been introduced to the lines with a one-cycle delay time of about 60 ms or longer.

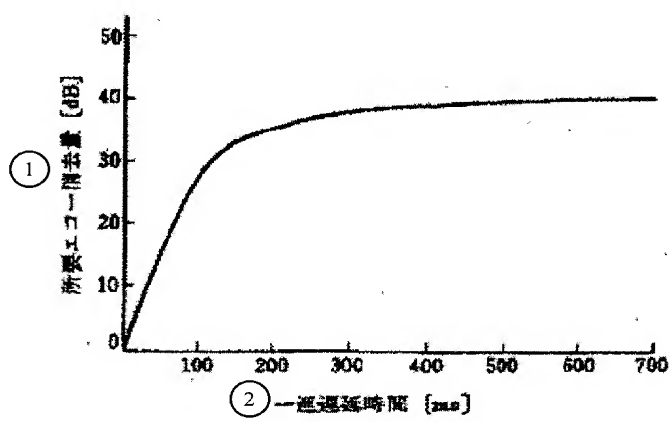


Figure 5.15 Required echo canceling quantity in line echo (telephone system).

Key: 1 Required echo canceling quantity
2 One-cycle delay time

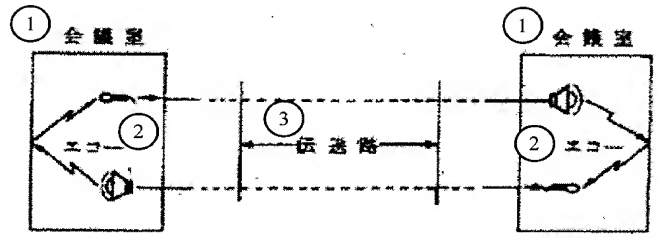


Figure 5.16 Structure of voice-amplified communication system.

Key:
1 Conference room
2 Echo
3 Transmission line

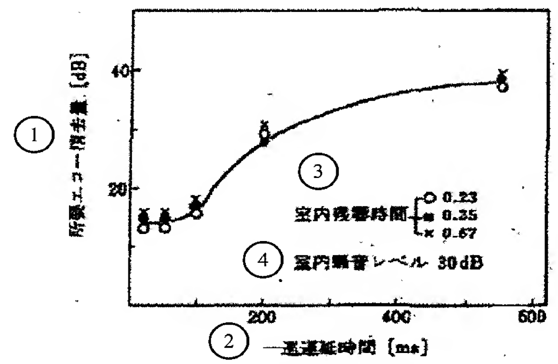


Figure 5.17 Required echo canceling quantity in acoustic echo (voice-amplified system).

Key:
1 Required echo canceling quantity
2 One-cycle delay time
3 In-room reverberation time
4 In-room noise level

Figure 5.16 shows the structure of a voice-amplified communication system using microphones and speakers. The voice of the caller is regenerated by a speaker on the caller side as echo after going through a microphone and with a certain delay time. Figure 5.17 shows the results of measurement obtained for the required echo canceling quantity (echo tolerable limit) versus the one-cycle delay time [24]. It differs from the case of line echo in that echo should be cancelled even if the one-cycle delay time is approximately tens of ms due to the influence of the acoustic echo path characteristics. Also, because the echo canceling time is the impulse response length including the absolute delay time from input to the speaker to output from the microphone, the echo canceling time is longer inside a room due to a longer reverberation time.

5.3.2 Structure and operation of echo canceller

The echo canceller operates as follows: the transmission characteristics of the echo path are estimated to form a replica of the impulse response, which is then superposed on the receiving signal to form a pseudo echo signal that is then subtracted from the true echo signal to cancel the echo. Because the transmission characteristics of the echo path vary over time, for the pseudo echo circuit, an adaptive type that determines the replica of the impulse response at all times is used, and, for the adaptive algorithm, real time operation, high speed and high precision are required.

Various structures have been proposed for echo cancellers [25]-[27]. Figure 5.18 shows a typical structure of an echo canceller. In the pseudo echo generating part, real-time operation can be performed, and high stability is guaranteed. In addition, a transversal type adaptive filter for which the method in estimating the transmission characteristics is established is often used [28], [29]. Also, various schemes for estimating the echo path have been proposed (for details, the reader is referred to Book 2, Chapter 10). Usually, a learning identification method (NLMS) that has a smaller computing quantity and allows real-time operation is adopted.

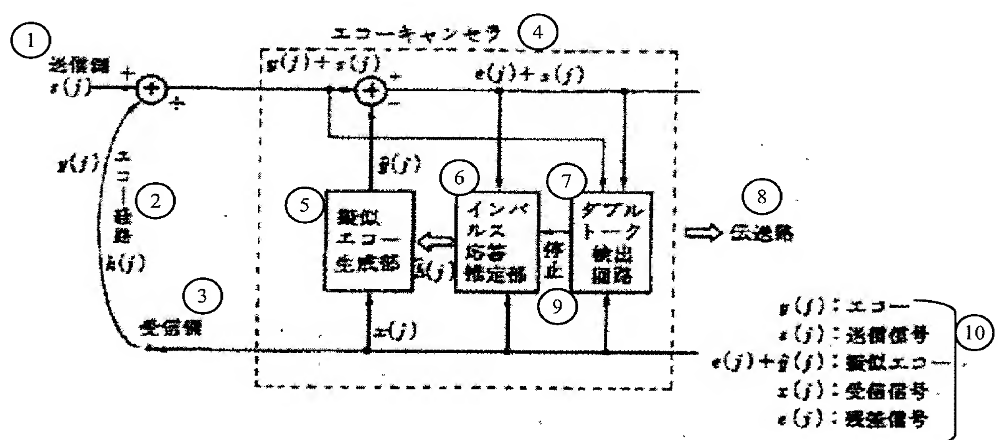


Figure 5.18 Structure of the echo canceller.

Key:
1 Sending side
2 Echo path
3 Receiving side
4 Echo canceller
5 Pseudo echo generating part
6 Impulse answer estimating part
7 Double-talk detector
8 Transmission line
9 Stop
10 Echo
Sending signal

Pseudo echo
Receiving signal
Residual signal

When echo canceling is to be realized, successive treatment is necessary with all of the arithmetic and logic operations finished within each sample, which requires a high arithmetic and logic operation clock speed that is proportional to the echo canceling time. For example, assuming that p represents the operation clock time (corresponding to the CPU operation processing time in DSP) and n represents the tap number, the product addition operation (convolution operation) becomes pxn, the operation of the correction quantity (depending on the algorithm) is pxnxC_1, the data transfer is pxnxC_2, the double torque control is pxC_3, and the sum of the treatment times for the arithmetic and logic operation becomes $s = p \times [n \times (C_1 + C_2 + 1) + C_3]$. Here, when the tap number n (corresponding to echo canceling time T, and n = T/T_s represents the sampling time) is increased, one has $s \approx p \times n \times k (k = C_1 + C_2)$. Also, the generally necessary memory number is $2 \times n + \alpha$. The first item refers to the memory required for the convolution operation, and the second item is for computing the correction coefficient. Consequently, when DSP is used, it is necessary to select a DSP that is appropriate for the echo canceller based on the operation processing speed and the memory capacity. Also, it is important to select possible types that allow subordinate connection of the DSP.

5.3.3 Echo canceller for long distance line

As a device for echo canceling, an echo suppressor has been used. Also, recently, an echo canceller free of a broken sensation in conversation has been introduced. For an echo canceller, in 1966, M. Sondhi of the ATT Bell Lab described an adaptive algorithm in theory [80]. However, with the analog technology available at that time, it was hard to realize a complicated adaptive algorithm and the echo canceling quantity demanded for an echo canceller.

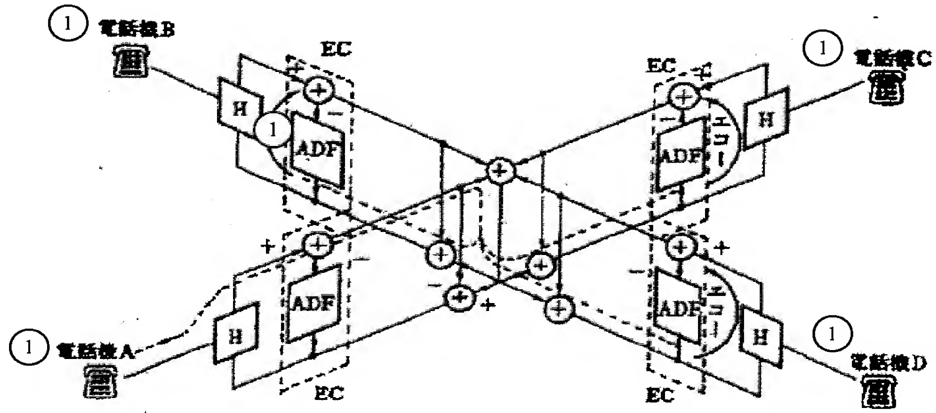


Figure 5.19 Generation of echo in telephone conference service (4 pairs connected to each other).

Key: 1 Telephone set ___

By the 1970s, with advent of digital signal processing technology that facilitated complicated and high precision arithmetic and logic operations and the rapid development of LSI technology, echo cancellers could be produced with small and low-cost hardware. The first echo canceller developed for satellite/domestic use was created by Duttweiller of the ATT Bell Lab., and the echo canceling time then was 8 ms (64 taps) [81]. Then, ATT and KDD developed a single-chip LSI with an echo canceling time of about 60 ms (corresponding to 480 taps). At present, echo cancellers are rapidly being introduced into satellite/domestic lines.

5.3.4 Echo canceller for telephone conference for persons at plural sites

Echo canceling technology is adopted not only in cases in which echo is simply cancelled, but also in treating various phenomena that take place due to echo, such as canceling hauling and noise signals.

With a telephone conference service that began in 1986, the system allows simultaneous connection among 30 pairs of unspecified subscribers [53]. Echo canceling technology is also adopted in this system. Figure 5.19 shows the state of generation of echo in the case of four pairs of connection among different sites. In the case of multi-pair connection, plural voices are added simultaneously, so that plural echoes are also added. Consequently, the total gain of the system is greater than one, and, in the worst case, hauling (shrieking) occurs, or quasi-hauling takes place. In this state, the voice of the conversation counterpart cannot be heard, and only hauling is heard. Consequently, by canceling the echo generated in a hybrid transformer corresponding to each subscriber, this phenomenon can be prevented. Usually, assuming that the number of the desired echo canceling pairs is n, the value should be 10·log n or larger [23].

5.3.5 Echo canceller for relaying received signal

For an analog telephone network, the subscriber information is stored in a local exchange, and most local exchanges are 2-wire type analog exchanges, and first, a common line signal system that transfers the control signal between exchanges must be introduced to it. Consequently, it is impossible to directly transmit a sending call to the destination, and an incoming call should be sent to the destination. Consequently, for an analog network, the transmission loss between the subscribers via the communication network increases, and a 2-wire line type bidirectional amplifier is needed for transmitting to the destination at a conventional volume. Figure 5.20 shows the structure of a bidirectional amplifier having the echo canceller introduced in it. An AGC for automatic compensation of the transmission loss is necessary. In addition, the loop gain in the 4-wire line region including the amplifier should be greater than one. Consequently, stable operation required increasing the AGC gain while having the AGC gain and the echo canceller canceling quantity cancel each other. At present, for a telephone line, the voice power in one direction can be compensated up to 24 dB [34].

5.3.6 Echo canceller for voice answering

In order to accommodate plural subscribers in a voice answering service, it is preferred that operation be performed with an exchange outside the city. Also, because a timer system or other voice answering service has a fixed-shape answering form, usually, an answer is given while an announcement has not finished. In this case, a voice signal and a push-phone multi-frequency signal, and other signals are sent as the answer signal. Consequently, on the receiving side of the trunk device of the out-of-town exchange, as described in Section 5.3.3, the voice answer announcement signal and the answer signal are mixed by echo and are received. Consequently, the SN deteriorates, and, in the worst case, the answer signal may be judged incorrectly. As a result, in order to improve the instant response to the answer and to improve the correct understanding rate, an echo canceller is used.

5.3.7 Acoustic echo canceller

(1) TV-conference

Usually, the reverberation time of a conventional conference room is in the range of 100-400 ms. When this is represented by a transversal filter with 8 kHz of sampling, it is about 4000 taps [35]. In addition, for a high quality voice band (7 kHz or higher), because the sampling frequency is 2-6 times higher, the tap number is huge.

When the acoustic echo is a direct wave from a speaker to a microphone, an indirect wave (so-called reverberation) caused by multiple reflections from the wall surfaces inside the room is superposed, and its impulse response is significantly different from the impulse response

of the line echo. People and objects move or the indoor temperature varies over time, so the characteristics also vary significantly [38]. Also, many noises, such as the noise of an air conditioner, voices from surrounding persons, etc., cause deterioration in the performance of the echo canceller. In addition, realization of the hardware is much more difficult than other types of echo cancellers.

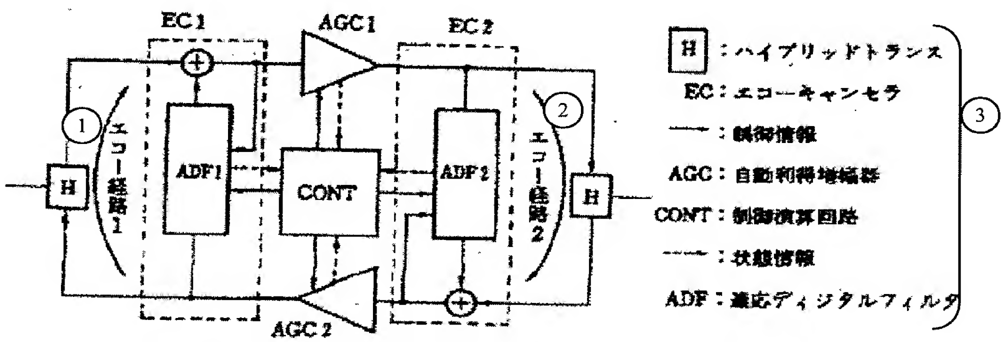


Figure 5.20 Structure of 2-wire bidirectional relay amplifier.

Key: 1 Echo path 1
2 Echo path 2
3 Hybrid transformer
Echo canceller
Control information
Automatic gain amplifier
Control operation circuit
State information
Adaptive digital filter

Consequently, it is possible to guarantee real-time operation by adopting the following methods: the tandem connection method [37], in which the necessary operation quantity and the storage capacity for each chip can be reduced by connecting plural chips in tandem and performing parallel treatment, and the band dividing method [38], in which the high band side is folded back to the low frequency side by means of modulation, so that the sampling frequency is lowered. Also, the convergence time that obtains the desired echo canceling quantity in the long tap length is longer in the learning identification method. Consequently, reduction in the convergence time by means of the following methods has been investigated: non-correlation method [39] that converts the input voice signal to a white signal, or plural band dividing method [40], and adaptive filter coefficient variation method [41] using the index attenuation characteristics of the impulse response.

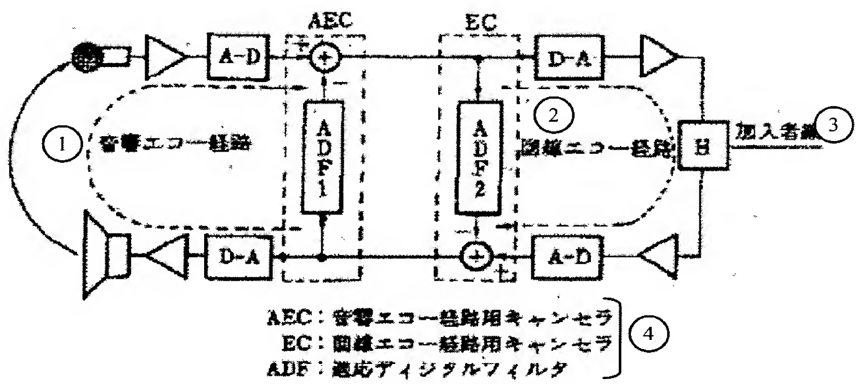


Figure 5.21 Structure of voice-amplified telephone using echo canceller.

Key:
1 Acoustic echo path
2 Line echo path
3 Subscriber line
4 Canceller for acoustic echo path
Canceller for line echo path
Adaptive digital filter

(2) Voice-amplified telephone (automobile telephone)

For a voice-amplified telephone with which calling is possible without a handset, due to the acoustic coupling between the speaker and the microphone, both acoustic echo and line echo due to a hybrid transformer are generated. For the voice-amplified telephone, in consideration of operability, the speaker and the microphone are usually integrated [42]. Because a low cost is necessary for a voice-amplified telephone, the tap number of the transversal type filter cannot be large. Consequently, it is used along with an echo compressor specifically to prevent the hauling phenomenon. Figure 5.21 shows an example of the structure of a voice-amplified telephone using an echo canceller.

Also, for a voice-amplified telephone, in order to guarantee safe driving, it is also adopted as a telephone for automobiles [43]. Since the space inside the automobile cabin is small, the attractive force of the sheet is high, so the reverberation time is shorter than in a conventional indoor environment, and the conversation bandwidth is 0.3-3.4 kHz. The hardware is smaller than that used in TV conferencing. However, the noise level is high during operation, and, due to the small space, movement of persons inside the automobile leads to variation in the transmission characteristics of the echo path. Consequently, the actual application environment is much more severe.

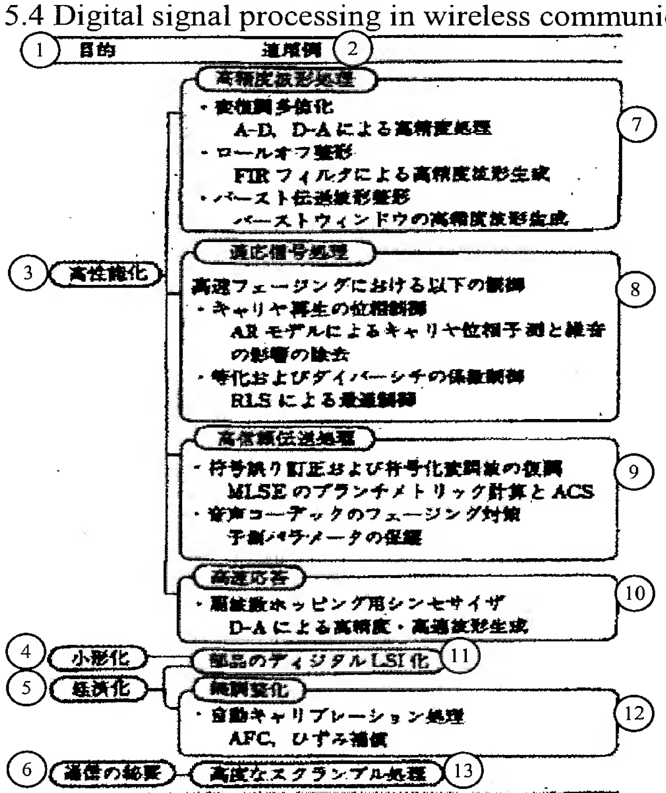
(M. Shimada)

5.4 Wireless communication

5.4.1 Application of digital signal processing in wireless communication

Typical wireless communication systems include mobile communication, stationary wireless communication, and satellite communication [44]-[47]. They have significantly different parameters and application conditions. Technical development has been from the viewpoint that the limited wireless frequency spectral resources should be efficiently used. In order to improve the performance and the reduce the size of the equipment, it is necessary to introduce digital signal processing (see Table 5.4). However, in applications in wireless communication, ① spurious transmission is strictly defined by the Electromagnetic Wave Law; ② the dynamic range of the receiving wave level is wide; ③ the carrier frequency of the transmission modulated wave is high,...[end of available text]

Table 5.4 Digital signal processing in wireless communication.



Key:

1	Purpose
2	Application examples
3	Higher performance
4	Smaller size
5	Lower price
6	Confidentiality of communication

7 High precision waveform processing
· Modem multi-value scheme
High precision treatment is performed by means of A-D, D-A conversion
· Roll-off wave shaping
High precision waveform generation by FIR filter
· Burst transmission waveform shaping
High precision waveform generation by burst window
8 The following controls are performed in fast phasing
· Phase control in carrier reproduction
Prediction of carrier phase by AR model and removal of noise influence
· Equalization and diversity coefficient control
Optimum control by RLS
9 High-reliability transmission processing
· Correction of code error and demodulation of encoded modulated wave
Branch metric computing of MLSE and ACS
· Phasing counter measure of voice codec
Protection of prediction parameter
10 Fast response
· Synthesizer for frequency hopping
High-precision, high-speed waveform generation by D-A conversion
11 Formation of digital LSI for members
12 [illegible] adjustment
· Automatic calibration treatment
AFC, distortion compensation
13 High-level scramble treatment

Digital Signal Processing Handbook

Published by Denshi Joho Tsushin Gakkai [IEICE] 1993
1st edition, 1st publication published on January 31, 1993
Editor: Denshi Joho Tsushin Gakkai [IEICE]
Address: 3-5-8 Shibakoen, Minato-ku, Tokyo 105 Japan
Phone: (03)3433-6691 (main) (Tokyo 2-35300)
Issued by Omu K.K.
Representative: M. Sato
Publishing House: Omu KK
3-1 Kimita-cho, Chio donku
Tokyo 101
Phone: (03) 3233-0641 (main)
(Tokyo 6-20018)

Printed by NBS in Japan
Book prepared by Samizusha

ISBN4-274-03423-2